THE SECRET GUIDE TO COMPUTERS

Not $9.95 or $19.95!
JUST $8!
We offer the BEST for LESS!

Used by:

CBS News, NBC, & ABC
The Wall Street Journal
The New York Times
USA Today
The Australian
others in 6 continents

Praised by:

Infoworld
Byte Magazine
Scientific American
Personal Computing
The Whole Earth Catalog
hundreds of others

Recommended by leaders of:

The Boston Computer Society
Nat. Council of Teachers of Math
Nat. Assn. of Independent Schools
The World Bank

Used at:

Harvard & M.I.T.
Polaroid & Gillette
Planters Peanuts & M&M's
many fire departments

VOLUME 1: SECRET SKILLS

All the popular magazines agree: *The Secret Guide to Computers* is the best computer book ever written. See the reviews on page 2. Clear, candid, and complete, each volume contains 9 power-packed chapters:

Volume 1: Secret Skills	Volume 2: Secret Thrills	Volume 3: Secret Chills
1 The computer industry	1 Amazing applications	1 Micro makers
2 The computer's parts	2 Electronic pleasures	2 FORTRAN
3 Buyer's guide	3 The literary computer	3 COBOL
4 Personal programming	4 Artificial intelligence	4 "C"
5 Powerful programming	5 Computerizing your business	5 Alternative languages
6 DOS	6 Advanced prog. using BASIC	6 Bit fiddling
7 Word processing	7 LOGO	7 Numerical analysis
8 Spreadsheets	8 PASCAL	8 Our past
9 Data-management sys.	9 DBASE	9 Your future

You have in your hands volume 1, which gives you these Secret Skills:

Computer jargon. Which manufacturers are best. How to get discounts. Tricks. How to program. Word processing. Choosing & using BUSINESS SOFTWARE. Tutorials in YOUR computer's BASIC, DOS, Word Perfect, 1-2-3, and PFS. Secrets in the IBM PC, Apple 2e & 2c, Mac, Commodore, Atari, & Tandy. Much more! See the GIGANTIC table of contents on page 6!

Whenever you have questions about computers, call the author, Russy-poo, at his home phone (617-666-2666), and he'll answer all your questions, FREE! Call 24 hours: he's usually in and sleeps only lightly. The press calls him AMERICA'S COMPUTER GURU. You too can call the guru, day or night, whenever you want free help. Take this SECRET BOOK home with you, and have a nice day!

Twelfth edition © 1986 by Russ Walter, just $8! To order, use coupon on back page.

Donated to
Internet Archive by
Kay Savetz

Dear You!

Congratulations! You've decided to learn about computers! That's reasonable—especially when you consider that computers all over the country are trying to learn about *you*.

This book will turn you into a computer expert, even if you're an idiot. In fact, *some of the world's most famous computer experts have read this book*—and are idiots!

This book tells you more about computers than your mom recommends. It explains the joys and sorrows, the highs and hassles, the excitement and the excrement. At times, you'll have a ball; at other times, you'll have a ball and chain. You'll giggle and gag, laugh and lament, frolic and frown. Parts of this book will please you so much that you'll shower me with tulips; other parts will make you shower me with tomatoes. But at least you won't be bored!

This is the only book ever published that's been extensively praised by *all* the popular computer magazines.

It's the only book that covers *all* aspects of computers. It's the only book that makes even the most advanced topics easy. And it's the only major book whose author is stupid enough to give out his home phone number.

I'm the author. My home phone number is 617-666-2666. Whenever you have a question about computers—or about life—call me, and I'll help you, free, even if your question is weird or personal. This free consulting service has saved readers many kilohours and kilobucks.

My phone number's easy to memorize: just dial the area code for Boston (617) and then, to figure out what-the-devil your computer's doing, dial the phone number for the devil's double: 666-2-666. Whenever your computer makes you ill, dial "Sick-sick-sick! Too sick-sick-sick!"

You can call 24 hours: I'm almost always in, and I sleep only lightly. Because of the midnight phone calls, I'm not married . . . unless you count the computers I've fallen in love with.

So for free help, phone. (Don't write.) Begin by saying your name, city, how you got my number ("from volume 1 of the 12th edition"), and a one-sentence summary of your question. Then I'll ask for details, and we'll have a friendly chat.

Whenever you visit the Boston area, drop by and say hello. You can use my extensive library, free, and also say hello to my 40 computers.

Drop by anytime: day or night! Phone first, to pick a time and get directions. I'm on Ashland Street, just a five-minute walk from the Porter Square stop on Boston's subway. (You buy your computers at Computerland, and we bury the bad ones here, at Ashland.)

You have in your hands volume 1. To learn even more about computers, read volumes 2 and 3. To order them, use the coupon on the back page.

Have fun with computers. But if someday you look at your heart and find it's turned into a microchip, don't bother blaming me: I've turned into a robot already!

At your service, your computer butler,

Russy-poo

"The butler did it."

P.S. This book was written by Russ's robots. (He thinks we're just "word processors", but we know better!) These dumb details will help your librarian fill in the file cards and not get fired.

Title: The Secret Guide to Computers, twelfth edition, volume 1
So-called author: Russ Walter (known to his friends as "Russy-poo")
Publisher: the same jerk as the author
Address: Russ hides in his bedroom at 22 Ashland St., Somerville, MA 02144
Copywrong: © 1986 by Russ Walter
Library of Congress number: "QA76.5 W286", until Congress changes its mind
International Standard Book Number: ha ha ha! You think this book is standard?

Staff: Russ Walter, Irene Vassos, Larry Mancini, Susan Goldenberg, Cindy Best. Consultants: Judy Hausman, Larry Govoni, Adam Green. Management alumnae: Priscilla Grogan, Cynthia Cobb. Printing: Russ Walter, John Pow, Semline.

PRAISED BY MAGAZINES

The Secret Guide to Computers has been praised by all the famous microcomputer magazines.

Personal Computing Magazine: "The Secret Guide to Computers is bulging with information, and you'll enjoy it. Russ Walter's approach to text-writing sets a new style that other authors might do well to follow. The Secret Guide is one of the most readable texts on computers that's appeared. It's readable, instructive, and downright entertaining. If more college texts were written in the Russ Walter style, more college students would reach their commencement day."

Popular Computing Magazine: "Russ Walter is king of the East Coast computer cognoscenti. His Secret Guide is the biggest bargain in computer tutorials in our hemisphere. If CBS ever decides to replace Andy Rooney with a '60 Minutes' computer pundit, they'd need to look no further than Russ Walter. His wry Walterian observations enliven nearly every page of his book. The Secret Guide is the first collection of computer writings that one might dare call literature. Walter couldn't have packed more practical tips into the Guide if he'd used a shoehorn, yet the writing remains graceful and witty, with freewheeling zaniness."

Infoworld: "Russ Walter is recognized and respected in many parts of the country as a knowledgeable and effective instructor. People call him at all hours of the night from as far away as Australia for advice. Talking to him is like talking to six people at once: he's an author, publisher, lecturer, teacher, consultant, and 'computer butler, at your service'. His Guide answers the questions he's asked most often. It's readable, outrageous, and includes a wealth of information."

Computer Update: "Russ Walter is Boston's computer guru. His Guide is cleverly graduated, outrageous, and funny. He turns computerese into plain speaking, while making you giggle. He's years ahead of the pack that claims to have ways of instructing computer novices. His Guide doesn't suffer from the oversights in manuals that unsuccessfully try to imitate his style. His unique mix of zany humor and step-by-step instruction avoids the condescending tone of manuals that attempt to follow his lead. His amusing Guide goes along with his courses, which are popular because of their unintimidating Walterian tone."

Eighty Micro: "Theatrical, madcap Russ Walter is a cult hero, one of the two pop preachers of the computer gospel. Though Peter McWilliams can write funny captions, Russ Walter's the one you'd call in the middle of a debugging session. Lots of books teach novices the difference between RAM and ROM, but Walter is more ambitious and more irreverent. He's superb at micro architecture. He has the best, most concise brand-name evaluations, which tell exactly what rookie shoppers should be told. This is probably the most intelligent beginner's book in the field."

Interface Age: "The Secret Guide claims to cover all the facets of computers simply but thoroughly, and to be foolproof, carefully tested and revised, fun, simple to read, and popular among kids and adults who want straightforward talk. The Guide meets all those claims. How did Russ Walter do it? His examples are simple, easy to understand, and cute. He teaches the fundamentals and much more. He's anticipated the problems that frustrate the beginner and minimizes them by careful instruction and reminders. His sense of humor, which ranges from corny to zany, makes the Guide fun. His Guide has gone through careful debugging and constant improvements, and shows his enthusiasm and experience. It's a best buy. It makes vivid, effective use of the English language, with clear examples, charts, and programs."

Softalk: "The Guide is funny and irreverent. The material is culled from special seminars given by Russ Walter, New England's computer guru. Many leaders of the computer industry are alumni of his courses, and his Guide is reminiscent of an old-fashioned medicine show: it has high energy, humor, and flair. He pulls no punches, and fires well-deserved salvos at many sacred cows. He's been updating the Guide for many years. It's long been a cult hit. He calls himself 'your computer butler'. This butler comes with some of the finest references in the industry, and you know how hard it is to find decent help these days."

Abacus: "Alternative-culture Walter provides the best current treatment of programming languages. His Secret Guide includes pungent, punchy descriptions of almost all the current languages. It's irreverent, reminiscent of the underground books of the 1960's. It's simple to read, fast-paced, surprisingly complete, full of locker-room computer gossip, and loaded with examples. Computing professionals could well recommend it to beginners. Walter takes pride in challenging and shocking the staid and comfortable."

Connecticut Computer Society News: "Russ Walter's books have been used by insiders for years. He's widely known as Boston's computer guru. He's special as a teacher because of three factors: his comprehensive knowledge of many computers and their languages, operating systems, and applications; his ability to break complicated processes into the smallest components; and his humor. A valuable feature of the Guide is his candid comments about various computers and software. He's one of the few people able to review languages, machines and software, all in a humorous, clear manner, with the whole endeavor set off by his sense of industry perspective, history, and culture. He gives out his home phone number, encourages people to call 24 hours a day, doesn't charge, and has several lines to handle all the calls. If you're ever struck with a computer problem, give Russ a call."

Scientific American: "This introduction to microcomputers and their culture is a Dayglow lifebuoy. It's irresistable. Every instruction leads to a useful result. Step by step the student gains control of language and reaches fluency. The Guide also introduces the rich culture of computers, everything from a page about Ramon Lull to the hyperbolic history of A.I. Walter's candor shines; he makes clear the faults and foibles others ignore or cast in vague hints. The effect is that of a private conversation with a well-informed talkative friend who knows the inside story. The text reads like the patter of a talented midnight disc jockey; it's flip, self-deprecatory, randy, and good-humored. His useful frank content and the coherence of his style make him unique. His detailed account of the workings of the false psychotherapist Eliza, a wonderful program that simulates Rogerian psychotherapy, is all by itself worth the cost of the volume. First-rate advice on what and how to buy are part of the rich mix. No room that holds a small computer and an adult learning to use it is well equipped without the Guide."

The Whole Earth Catalog, in its Coevolution Quarterly update: "Though the personal-computer subculture continues to explode, it's still kept some of the fierce honesty for which it was noted in its early years. The Secret Guide is one of the few introductory books to carry on that tradition in print, and the only introductory survey of equipment that's kept up to date. Russ Walter jokes, bitches, enthuses, condemns, and charms. He answers readers' questions for free. He's kept that up for years without burning out. The book tells the bald truth in comprehensible language."

The Detroit News: "Russ Walter is a legendary teacher of personal computers. He wears roller skates and a witch's costume (complete with funny hat) and answers the phone any time of the day or night. His Secret Guide is fiercely honest, and it packs an incredible amount of information. Although lots of books claim to be comprehensive, his Secret Guide is the only one that really includes everything. He gives you all the dirt about the companies and their hardware and even evaluates their business practices. He finds out about problems that computer companies try to keep to themselves, and he unflinchingly exposes them. He also compares and evaluates every computer language this reviewer has ever heard of. The next time you're stumped with a balky computer, pick up the phone and call that expert. You'll always get a truthful answer."

The Australian, one of Australia's major periodicals: "Read the Secret Guide and you'll have a very good idea of how computers and programs work—and a good laugh as well. The blurb says the book turns computerese into plain English and shows computers aren't boring; for once, a blurb does not lie. The book's coverage of programming is intelligent, urbane, and extremely funny. It's full of great programming and publishing ideas. Plainly, Russ Walter has a few kangaroos loose in his top paddock, because he says to call 24 hours even though the midnight phone calls have prevented him from getting married except to computers. Yes, he does answer the phone and is delighted to answer any questions."

Cider Press: "The Guide is as difficult to lay aside as a first-rate mystery. It unravels the computer's mysteries effortlessly and humorously. The Guide should be given to all beginners with the purchase of their computers."

Computer Bargain Info: "Russ Walter is Boston's humble computer guru. His Secret Guide is widely acclaimed by experts as brilliant. For further pleasures, 24 hours, call Russ."

Enter: "The Secret Guide is the best book about the different computer languages."

Compute: "Russ Walter is an industry leader."

Classroom Computer Learning: "Russ Walter's courses are intensive and inexpensive."

Barron's: "Russ Walter is a computer expert who answers consumers' questions for free and has been inundated by calls."

Many other periodicals have offered similar praises. Byte Magazine recommends that computerists get Russ Walter's advice. Microcomputing Magazine says, "Plan ahead: get in on the secret now." Creative Computing calls the Guide "fascinating, easy to understand, an excellent book at a ridiculously low price" and "especially endorses" the Guide. The Wall Street Journal, on the front page of its second section, printed Russ's home phone number and called him "a computer expert, a guru who doesn't mind phone calls. He fields questions from consumers, and invites calls about computers day or night." Esquire recommends the Guide over all other computer books and says it "contains lots of fact and opinion untainted by bias." The Boston Globe, The Philadelphia Inquirer, and many other newspapers, TV stations, and radio stations have quoted Russ extensively and called him "a computer expert".

FAN MAIL

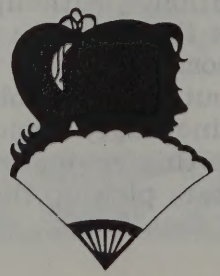
From our readers, we've received *thousands* of letters and phone calls, praising us. For example, here are the first four we received. . . .

Head of a meat factory: "I spent $100 on books on BASIC. Yours was the only one I could understand."

Head of a computer center: "Terrific! I've been looking for good books on BASIC, and yours is as thorough as the most complete but as readable as the most elementary. Yours is better than any of the others. You have them beat by a mile."

Head of a large publishing company: "I must tell you how much I enjoyed reading your computer books. This is stranger than you'd imagine, for I find the machines deplorable. But you've taken such a refreshing, comical, and interesting approach, even the likes of me can enjoy them."

Head engineer at the atomic-weapon design headquarters: "I have a big desk piled high with a lot of publications that talk a lot but don't say anything. Your book is an exception. The Secret Guide has assumed the importance here of a very critical top-secret document. It will be required reading for the whole crew."



INTERVIEW

In this interview, Russy-poo answers the most popular questions about this book and what's behind it.

Why did you write The Secret Guide to Computers? I saw my students spending too much effort taking notes, so I made up my own notes to hand them. Over the years, my notes got longer, so that the Guide's become a three-volume set whose twelfth edition totals about 900 pages. Each time I develop a new edition, I try to make it the kind of book I wish I had when *I* was a student.

Why do you charge less than half of what other publishers charge? I'm not trying to make a profit. I'm just trying to make people happy--by charging as little as possible, while still covering my expenses. Instead of "charging as much as the market will bear", I try to "charge so little that the public will cheer."

What does the Guide cover? Everything. Every computer topic is touched on, and the most important topics are covered in depth.

Which hardware and software does the Guide recommend? For the "typical" consumer, it recommends either an IBM Personal Computer or an imitation, such as the *Leading Edge Model D*. For a printer, it recommends the *Panasonic 1091*; for word processing, it recommends *Word Perfect 4.1*; for spreadsheets, it recommends *1-2-3* or *Excel* or an imitation; for data management, it recommends *Q&A*. It recommends buying software from a good discount dealer, such as *Logicsoft*. But it also recommends alternative equipment, software, and suppliers, in case your own needs are not "typical". For example, it says under what conditions you should choose computers by Apple, Commodore, Radio Shack, Atari, and other companies instead. And since new products, bargains, and improvements come out every week, you should phone me for the latest hot tips, before you buy anything.

Do you really answer the phone 24 hours a day? When do you sleep? When folks call in the middle of the night, I wake up, answer their questions, then go back to bed. I'm near the phone about 85% of the time, though occasionally I have to go out on an errand. If you get no answer, I'm out briefly, so please call again; if you get an answering machine, I'm out on a longer project, so leave your name and number and I'll call you back at *my* expense, even if it's long distance.

Why do you give the phone help for free? Are you a masochist, a saint, or a nut? I give the help free for three reasons: I like to be a nice guy; it helps me keep in touch with my readers, so I can get suggestions on how to improve the Guide further; and it encourages people to tell their friends about me, so that I don't have to spend money on advertising.

At computer shows, do you really appear as a witch? I wear a witch's hat, with a monk's habit and roller skates. All of them are shiny black, for contrast with my white gloves. Why? Because it's fun!

Did you write the whole Guide yourself? Yes. But I received many suggestions from my readers, friends, and staff, and some of the examples and phrasings were contributed by Irene Vassos and Coleco.

What's your background? I hold degrees in math and education from Dartmouth and Harvard, I've been on the faculty of several colleges (Northeastern and Wesleyan and Wellesley), and I was one of the founding editors of Personal Computing Magazine. But most of my computer expertise comes from spending many hours every day reading computer books and magazines, discussing computer questions on the phone, and analyzing the philosophical issues that underlie all those details.

Who reads the Guide? All sorts of people. Kids read it because it's easy to understand; computer professionals read it because it contains lots of secret tidbits you can't find anywhere else.

Besides publishing the Guide and answering the phone, what else do you do? I take my 40 computers to many cities, where I give intensive computer courses that are almost free. I've also been writing another book, called The Secret Guide to Tricky Living, which covers everything *except* computers!

TABLE OF CONTENTS

★★★★★★★★★★★★★★★★ CONTENTS OF VOLUME 1: SECRET SKILLS ★★★★★★★★★★★★★★★★

Chapters 1-3 reveal how computer systems work, the secrets of the companies that developed them, which brands to avoid, which brands to buy, and how to get discounts. We save you money and grief.

Chapters 4-5 turn you into an amazing programmer, who uses the BASIC language as a sword to cut through all of life's problems. You'll have fun. (Many families can't sleep, because they spend the whole night giggling at these chapters.)

Chapters 6-9 train you in using the most popular software for the IBM Personal Computer and competitors, so that you become a guru of DOS, word processing, spreadsheets, and data-management systems. You get extensively tutored in PC-DOS (and MS-DOS), Word Perfect (versions 4.0 and 4.1), Writing Assistant (versions 1.00 and 1.01), Lotus 1-2-3 (releases 1A and 2), and PFS File (version C), then learn about their competitors, so that you can accurately pick the software that's best for *you*.

Appendix 1 explains how *your* computer differs from the standard. It explains *your* computer's keyboard, disk system, and version of BASIC. It uncovers the peculiarities of the IBM PC, Commodore 64, Atari 800 XL, Apple 2c & Macintosh, TRS-80 Model 4 & Color Computer, and many other famous computers. It shows how to convert programs from alternative computers to *your* computer.

Appendix 2 prevents you from ever getting lost. It displays all the lists and indexes you need, to locate information quickly. It also contains a coupon so that you get volumes 2 & 3, intensive courses, and brochures, at little or no charge.

Here are the details. . . .

CHAPTER 1: THE COMPUTER INDUSTRY

Kinds of computers (sizes, maxi, mini, micro, choose your size, styles) 12
Four houses (the house that Jack built, Steve's, trash from Texas, IBM) 16

CHAPTER 2: THE COMPUTER'S PARTS

Screens (what the screen shows, TV, monitors, video terminals, LCD) 22
Chips (main circuitry, CPU, memory fundamentals, RAM, ROM, mail order) 26
Disks (appearance, when to use, floppies, buying, protecting, hard) 34
Tapes (tapes versus disks, cassettes, big reels) 42
Printers (fundamentals, quality, feed, speed, width, interfacing, graphics) 44
Exotic I/O devices (input versus output, speakers, graphics input, modems) 51
Software (wares, DOS, languages, programs, applications, data disks, paying) 53

CHAPTER 3: BUYER'S GUIDE

Subculture (drugs, my marriage, your marriage, why buy a computer) 58
Periodicals (why read, how to buy, reviews, newspapers, magazines) 62
Discount dealers (how to, the famous, Jews, Tandy, retail, shows, used, wary) 65
IBM & clones (PC, Models D & M, Junior, XT, AT, Tandy, Compaq, others) 72
Apple (Apple 2e & 2c, Apple's expensive computers, Apple 3, Lisa, Macintosh) 80
Commodore (Commodore 64, Vic-20, black computers, Commodore 128, Amiga) 84
Alternative computers (Atari, Coco, pocket, lap, CP/M, Tandy's Z-80, Adam) 87

CHAPTER 4: PERSONAL PROGRAMMING

Entering BASIC (BASIC, keyboard, starting, math, strings, long commands) 92
Easy programs (how to program, mistakes, rearranging, tens, ranges, colons) 100
Tricky printing (semicolons, spaces, multiple calculations, clearing, on paper) 108
Going & stopping (GO TO, life's a loop, infinite pause, skipping, END, STOP) 112
Variables (numeric variables, when to use, string variables, long names) 118
INPUT (string input, admissions, vs. PRINT, stories, bills, numeric, convert) 122
IF...THEN (therapist, keywords, Mary Poppins, the strange & fancy, ELSE) 128
Aids (reminders, disks, tapes, debug, edit, renumber, pause, apostrophe) 136

CHAPTER 5: POWERFUL PROGRAMMING

DATA...READ (lists of data, Henry the Eighth, party, debts, diets) 144
FOR...NEXT (counting, meetings & squares, midnight, color, finite pause) 154
Pretty output (zones, TAB, calendar, pixels, sounds) 163
Elements (constants, kinds of variables, expressions, statements, boxes) 174
Loop techniques (increments, decs, counts, quiz, sums, checks, series, data) 177
Random numbers (prediction, integers, games, friends, horoscopes, decimals) 185

CHAPTER 6: DOS

Getting into DOS (IBM PC software, first disk, versions, starting, directory) 192
External commands (SORT, CHKDSK, absence, BASIC, DISKCOPY, FORMAT) 196
Internal commands (popular commands, main drive, summary, 2 ways to copy) 200
Advanced features (printer, batch files, boot-up, copyrights, subdirectories) 203

CHAPTER 7: WORD PROCESSING

Background (what is it, how invented, 3 definitions, the 2 best systems) 208
Quick tricks (starting, typing, moving cursor, deleting, replacing, inserting) 211
Use your template (what it means, underlining, boldfacing, centering, blocks) 214
Final steps (your last 3 steps, Writing Assistant's wrap-up, Word Perfect's) 217
Advanced features (margin, header, tab, search, spell, add, macro, driver) 220
What to buy (word processors for the IBM PC, for other computers) 223

CHAPTER 8: SPREADSHEETS

Bop around (kinds, starting 1-2-3, mistakes, functions, point, center, fill) 228
Slash your problems (/WEY, tricks, /RE, /WD, /WI, MOVE, COPY) 233
Big sheets (how big, extra rows & columns, width, titles, windows, manual) 237
Final steps (format the numbers, sort the data, print on paper, save, quit) 241
Graphs (simple graphs, multiple, labeling, bar, other types, finishing, paper) 244

CHAPTER 9: DATA-MANAGEMENT SYSTEMS

Concepts (what's data management, file jargon, sample, vs. word processing) 250
PFS File (what is it, build, search, keyboard, calculate, mass, layout, exit) 253
Beyond PFS File (PFS Report, PC-File, Reflex, Q&A, DBASE & other DBMS's) 264

APPENDIX 1: VERSIONS OF BASIC

IBM PC 270
Commodore 64 275
Atari 800 XL 282
Apple 2c 290
Apple Macintosh 300
TRS-80 Color Computer 2 308
TRS-80 Model 4 314
Other computers 318

APPENDIX 2: LISTS

Themes 328
Commands 329
BASIC examples 330
Error messages 332
Coupon 334

★★★★★★★★★★★★★★★★ CONTENTS OF VOLUME 2: SECRET THRILLS ★★★★★★★★★★★★★★★★

Chapters 1-5 shower you with novel ways to use the computer: you'll see Amazing Applications, enjoy Electronic Pleasures, create The Literary Computer and Artificial Intelligence, and learn the best way of Computerizing Your Business. Chapter 6 reveals the tricks of Advanced Programming using BASIC. Chapters 7-9 take you beyond BASIC, by making you fluent in three other popular languages: LOGO, PASCAL, and DBASE.

★★★★★★★★★★★★★★★★ CONTENTS OF VOLUME 3: SECRET CHILLS ★★★★★★★★★★★★★★★★

Chapter 1 chronicles the ridiculous rise of IBM, Apple, Commodore, Radio Shack, and Atari, and the sad fall of some of those companies and other famous Micro Makers. Chapters 2-7 turn you into a technical whiz at FORTRAN, COBOL, C, Alternative Languages, Bit Fiddling, and Numerical Analysis. Chapters 8-9 help you make wise decisions about getting ahead, by analyzing Our Past and Your Future.

SPECIAL SERVICES

FREE REPRINTS

You may reprint up to 10 pages of The Secret Guide to Computers free, and make as many copies of them as you like. Just phone me, say which pages you're reprinting, and put this notice at the beginning of your reprint:

Most of this material comes from the twelfth edition of The Secret Guide to Computers, *copyright 198o by Russ Walter and reprinted with permission. To get* FREE LITERATURE *about the complete Secret Guide to Computers, phone Russ at 617-666-2666, 24 hours (he's almost always in); or send a postcard to him at 22 Ashland St., Somerville, MA 02144.*

Then send me two copies of your reprint.

If you want to reprint *more* than 10 pages, there's typically a charge, but it's tiny. Phone me first, to find out how many pennies.

Give the reprints to your friends and colleagues—or sell them to strangers. The Guide has been reprinted by many teachers (for their students), consultants (for their clients), stores (for their customers), magazines (for use as articles), and book publishers (for use in *their* books). You may even insert your own comments, put your own name on the cover, call yourself a co-author, and become famous!

PRESERVED CLASSICS

You're reading the 12th edition. We've managed to preserve a few copies of the previous editions, which were more fun and contained more details about the classic computers. We can give you the 11th edition for just $4 per volume, the 10th for just $2 per volume, and the 9th for just $1 per volume. They're ideal for schools on tight budgets and for low-cost gifts to your friends. To find out what mysteries lurk in those devilish old volumes, phone.

Some of those classic editions have been reprinted by Bates College, Scelbi Publications, Brady Publishing, Birkhauser Boston Publishing, and Coleco. But you can get the classics less expensively by phoning me directly.

BLITZ COURSES

I give courses to public and private groups, all over the world. The courses are almost free. My favorite is the weekend "blitz", which turns you into a complete computer expert in one weekend, by giving you many *dozens* of computers, a highly trained team of consultants, and this intensive schedule. . . .

Saturday	Sunday
9AM-11AM: how to buy hardware	9AM-11AM: advanced programming
11AM-1PM: programming in BASIC	11AM-1PM: software & hardware hassles
1PM-2PM: lunch & tutoring	1PM-2PM: lunch & perspectives
2PM-4PM: DOS & word processing	2PM-4PM: languages beyond BASIC
4PM-6PM: spreadsheets & databases	4PM-5PM: careers & your future
6PM-7PM: dinner & consulting	
7PM-9PM: arts & education	

The course costs just $2.75 per hour for individuals, and even less if you get a group rate with your friends. For details, use the coupon on the back page, or phone.

STRANGE STUFF

If you get together with your friends and buy many copies of the Secret Guide directly from us, we give you large discounts, so that you pay hardly anything. To find out about the next edition, The Secret Guide to Tricky Living, and videotapes, get on our mailing list, by using the coupon on the back page. Phone 617-666-2666 whenever you have questions about computers or life: I've been answering personal questions about everything from sex to skunks.

My editor told me to put this page in.
You don't have to read it.

DEDICATION

I dedicate this book to the computer, without whom I'd be unemployed.

WHAT THIS BOOK WILL DO FOR YOU

This book will make you even richer than the author! Unfortunately, he's broke.

PREREQUISITE

This book was written for idiots. To see whether you can get through the math, take this test: try to count to ten, but (here's the catch!) without looking at your fingers. To remove the temptation, cut them off.

ACKNOWLEDGMENT

I'd like to thank:

my many friends, whose names I've gladly forgotten;
my students, who naturally aren't my friends;
my typewriter, which has a mind of its own;
all others who helped make this book impossible.

APOLOGY

Any original ideas in this book are errors.

DISCLAIMER

The author denies any knowledge of the scintillating illegal activities he depicts.

COPYWRONG

Reproducing this material in whole or even in itsy-bitsies, without written acknowledgment of the author's hex sign, is a no-no, punishable by death, or at least by salt in your orange juice on April First. If you're a reviewer, you may quote this wonderful book, but not if you're nasty.

FORWARD

. . . because it's too late to turn back.

WHO ARE WE?

ABOUT THE SO-CALLED AUTHOR

The author ("Russy-poo" Walter) and the modern computer ("stored-program computer") were both conceived in 1946. Nine months later, Russy-poo was hatched; the modern computer took a few years longer. So Russ got a head start. But the computer quickly caught up. Ever since, they've been racing against each other, to see who's smartest.

The race is close, because Russ and the computer have so much in common. Impartial observers say the computer "acts human" and say Russ's personality is "as dead as a computer".

Yes, Russ resembles a computer in many ways. For example, both are Jewish. The father of the modern computer was John von Neumann, a Jew of German descent who, after living in Hungary, fled the Nazis and became a famous U.S. mathematician. The father of Russy-poo Walter was Henry Walter, a German Jew who fled the Nazis and became a famous U.S. dental salesman: to dentists, he sold teeth, dental chairs, and balloons (for the kids).

To try to get ahead of the computer, Russ got his bachelor's degree in math from Dartmouth in yummy '69 and has sadly remained a bachelor ever since (unless you count the computer he got married to). After Dartmouth, he got an M.A.T. in mathematical education from Harvard. Since he went to Harvard, you know he's a genius. Like most genii, he achieved the high honor of being a junior-high teacher.

After his classes showered him with the Paper Airplane Award, he moved on, to teach at an exclusive private school, for girls who were *very* exclusive. ("Exclusive" means everyone can come except you.)

After teaching every grade from 2 through 12 (he taught the 2nd-grade girls how to run the computer, and the 12th graders less intellectual things), he fled reality by joining Wesleyan University's math Ph.D. program in Connecticut's Middletown (the middle of Nowhere), where, after a year-and-a-half of highbrow hoopla, he was seduced by a computer, to whom he's now happily married.

After the wedding, he moved with his electrifying wife to Northeastern University in Boston (home of the bean and the cod), where he did a hilarious job of teaching in notorious Department of "Graphic Science". After quitting Northeastern and also editorship of *Personal Computing*, he spends his time now happily losing money by publishing this book.

Because his wife was lonely, Russ bought her 40 computers to keep her company, with names such as "Anita Atari", "Aphrodite the Apple", "Baby Blue Burping Bonnie", "Coco the Incredible Clown", "Jack the Shack", "Kooky Casio", "Slick Vic", and "Terrible Tina with her Texas Instruments". He hides them in a van and they drive around the country having parties with anyone who wishes. When Russ's van comes to your town, he'll tell your mayor that the van is just for "educational purposes"; but what an education! His groupies all strip off their exteriors, to teach students a thing or two about computer anatomy.

Banned in Boston, Russ and his groupies moved north, where they hide in a pleasure palace underneath the Porter Square pine tree. Each room in the palace has a nickname. Come visit the "Input Room" (kitchen), bathroom ("Output"), three high-tech rooms ("Production", "Research", and "Creativity"), and four devilish rooms ("Sunshine", "The Cavern", "Pleasure", and "Pain").

As for Russ himself, here are his statistics, from head to toe. . . .

Head: *in the clouds* **Hair**: *departing* **Brow**: *beaten*
Eyes: *glazed* **Lashes**: *40* **Nose**: *to the grindstone*
Mouth: *big* **Smile**: *bionic* **Tongue**: *should be bitten* **Teeth**: *remembered*
Cheeks: *blushing* **Chin**: *up* **Shoulders**: *burdened* **Wrists**: *watched*
Hands: *some* **Thumbs**: *up* **Ring finger**: *naked* **Heart**: *all*
Back: *everybody's on it* **Ass**: *an accurate description* **Buns**: *toasted*
Knees: *knocked* **Heeled**: *well* **Arches**: *gothic* **Toes**: *stepped on*

He wears a stuffed shirt, slick slacks, and sacramental socks—very holy! Here's his resumé. . . .

Age: *too* **Sex**: *yes!* **Race**: *rat* **Religion**: *Reformed Nerd*
Address: *wear pants instead* **State**: *distressed*
Father: *time* **Mother**: *earth* **Spouse**: *Brussels*
Car make: *never made it in a car* **License**: *poetic*
Occupation: *vegetable* **Career goal**: *play dead* **Hobbies**: *sleeping & crying*
Sports: *dodging tomatoes* **Greatest pleasure**: *hiding under the sink*
Favorite food: *thought* **Humor**: *less*

ABOUT THE COMPANY

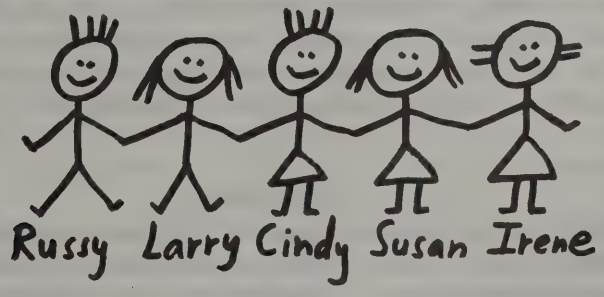
What company? C'mon over, bring a bottle of booze, and then we'll have some helluva company!

Come visit our Home Office, which is in Russ's home. Come visit our staff. We all have nicknames that begin with R:

Russ himself *(whose friends call him "Russy-poo")*
Reeny *(Irene Vassos, feminist farmer who loves Z!, Zimmer Bradley, and zucchini)*
the Rocker *(Larry Mancini, the gigabyte guitarist who loves gigs and bytes)*
the Religious *(Susan Goldenberg, the joyous Jew who telexes her prayers to God)*
the Radio *(C.B., Cindy Best, our "Best" employee)*

If you want to join our staff, the first requirement is to describe yourself by using the letter R.

This portrait of our staff was drawn by the internationally famous founder of the Kinderkook School of Art:



WOW! YOU FINISHED THE OPENING COMMENTS!
TURN THE PAGE FOR CHAPTER 1: THE COMPUTER INDUSTRY.

KINDS OF COMPUTERS

SIZES

Computers, like people, come in three sizes: you can get big ones, little ones, and pip-squeaks. The big'uns are called *maxicomputers* (or *mainframes*); little'uns are called *minicomputers*; and the teeny-weeny itty-bitty pip-squeaks are called *microcomputers*.

But how big is big? How little is little? How teeny-weeny is teeny-weeny? Opinions differ. It's like trying to measure a person. What's a "big" person? It could mean three things:

You're tall.
You're fat.
An interesting part of you is large.

The same is true for computers.

Yes, the word "maxicomputer", like the word "big", is vague. But here are some guidelines. . . .

Usually, a maxicomputer fills a room; a minicomputer fits in a corner; and a microcomputer freaks out on a desktop. Usually, a maxicomputer costs anywhere from $10,000,000 down to $100,000; a minicomputer costs from $100,000 down to $10,000; and a microcomputer costs from $10,000 down to $1. Yes, even *you* can afford some sort of microcomputer! If you're lucky, you'll even get one as a birthday present, from your Mommy or Daddy or Hubby or Wifey.

The largest maxicomputers are called *supercomputers*: they cost about $10,000,000, or even slightly more. The largest minicomputers are called *superminicomputers*: they cost about $100,000, or even slightly more. The largest microcomputers are called *supermicrocomputers*: they cost about $10,000, or even slightly more.

If a computer costs about $100,000, the salesperson will call it "a superminicomputer", or "the world's most powerful minicomputer", or "the world's least expensive maxicomputer", or—better yet—"the only computer that gives maxicomputer power at a minicomputer price". Now that you know how to sell computers, go sell lots of them, make lots of money, and please give me a cut!

Maxicomputers were invented long ago; minicomputers came more recently; microcomputers are ultra-modern. Companies began selling maxicomputers in the 1950's, minicomputers in the 1960's, and microcomputers in the 1970's. Today, you can buy all three sizes.

MAXICOMPUTERS

The dominant manufacturer of maxicomputers is *IBM*, which stands for *International Business Machines Corporation*. Too often, it also stands for "Incredibly Boring Machines", "Inertia Breeds Mediocrity", "International Big Mother", "Imperialism By Marketing", "Intolerant of Beards & Moustaches", "Idolized By Management", "Incompetents Becoming Managers", "Itsa Big Mess", and "It's Better Manually".

Traditionally, IBM computers have been colored blue, so that IBM has been nicknamed "Big Blue".

Why is Big Blue the dominant computer company? That's remained a mystery to many young programmers, who ask, "Mommy, why is the sky Big Blue?"

IBM's newest maxicomputers have boring names: they're called the *IBM 3080 series*, the *IBM 3090 series*, and the *IBM 4300 series*.

The other popular maxicomputer manufacturers are:

Burroughs (whose computers are called "burritos")
Univac (a division of Sperry)
NCR (the new, high-tech name for the ancient National Cash Register Company)
Control Data Corporation (CDC)
Honeywell (whose original factory was next to a well)

Since the initials of those companies are B, U, N, C, and H, that group of companies is called the *Bunch*. Computer analysts say, "Most maxicomputers are made by IBM and the Bunch."

MINICOMPUTERS

All the companies that make maxicomputers also make smaller computers—minicomputers. But the most fascinating minicomputers are made by companies that *specialize* in minicomputers.

The most popular minicomputers are the *PDP-11* and the *VAX-11*, which are made by a Massachusetts minicomputer specialist called *Digital Equipment Corporation (DEC)*.

Other popular minicomputer specialists—also in Massachusetts—are *Data General (DG)*, *Prime* (jokes about which are called "Prime ribs"), and *Wang* (founded by Dr. An Wang, a Chinese immigrant who's become the richest man in Massachusetts: together with his wife and children, he's worth about a billion dollars).

The only important minicomputer specialist that's *not* in Massachusetts is a California company called *HP*, which stands for *Hewlett-Packard* and "high-priced".

MICROCOMPUTERS

The most popular microcomputers are made by IBM—the company which is the "top dog"—and a group of companies called the *CAT*, which consists of *C*ommodore, *A*pple, and *T*andy. The microcomputer industry is a fight between the top DOG (IBM), the CAT, and hundreds of less successful competitors, who are called MICE, because they scurry about in perpetual fear of the DOG and the CAT.

Microcomputers fall into four price ranges: "almost free" ($1-$50), "cheap" ($50-$500), "mid-range" ($500-$1500), and "expensive" ($1500-$10000).

Of the microcomputers that are "almost free", the most popular ones hide inside digital watches and pocket calculators. They cost about $2 to manufacture.

The most popular "cheap" microcomputer is the *Commodore 64*, which costs about $150. You can buy it at toy stores (such as Toys-R-Us and Child World) and department stores (such as Sears and K-Mart).

The most popular "mid-range" microcomputers are made by Apple: they're called the *Apple 2 extended (Apple 2e)* and the *Apple 2 compact (Apple 2c)*. Each costs about $800. You buy them at fancy computer specialty stores, such as *Computerland*.

The most popular "expensive" microcomputer is the *IBM Personal Computer (IBM PC)*, which costs about $2300. It's sold at IBM's own stores (called *IBM Product Centers*) and other fancy stores (such as Computerland). It's also sold by IBM's *national-accounts sales force*, which visits the managers of America's largest companies and convinces them to give IBM computers to all employees.

IBM has also invented a portable version (the *Portable PC*), a stripped-down version (the *PC Junior*), a souped-up version (the *PC XT*, which stands for "PC Xtended"), and an even fancier version (the *PC AT*, which stands for "PC with Advanced Technology").

Many of the MICE make imitations of the IBM PC. Those imitations are called *PC compatibles* or *PC clones*. Some of the PC clones are so ridiculous that IBM laughs at them and calls them *PC clowns*.

Besides the IBM PC and clones, another popular expensive microcomputer is the *Macintosh*, which is made by Apple.

The company making the greatest variety of microcomputers is Tandy. It's the only company making microcomputers in all four price ranges ("almost free", "cheap", "mid-range", and "expensive"). Tandy sells them only in Radio Shack stores.

Most Radio Shack stores are owned by Tandy and charge full list price, but a few Radio Shack stores are independently owned franchises that give generous discounts.

Computers sold by Tandy are called *Tandy's Radio Shack computers (TRS computers)*.

CHOOSE YOUR SIZE

To computerize your home or small office, buy a microcomputer: it's all you can afford!

To computerize a larger company, buy a maxicomputer, or a few minicomputers, or lots of microcomputers.

The Secret Guide to Computers emphasizes microcomputers, for four reasons:

Most computers are microcomputers.

Unless you're rich, a microcomputer is all you can afford.

Even if your business is big enough to afford a larger computer, you'll probably get better service—and also save money—by wiring lots of microcomputers together instead.

Because microcomputers were invented more recently than larger computers, microcomputers tend to be more modern and more responsive to the needs of modern society.

Although the Guide emphasizes microcomputers, it discusses larger computers also, because *some* large businesses require them.

Every year, computer prices drop about 30%! So the longer you wait to buy a computer, the less you'll pay. But the longer you wait, the longer you'll be deprived of having a computer, and the further behind you'll be in computerizing your life and turning yourself into a computer expert.

Don't wait. Begin your new computerized life *now*!

STYLES

Computers come in six popular styles: *hidden*, *pocket*, *lap*, *TV*, *full-monitor*, and *multi-user*.

A *hidden computer* hides inside another device. For example, a computer hides inside your digital watch; another computer hides inside your pocket calculator; another computer hides inside your video-game machine. Since such a computer spends its entire life dedicated to performing just one task (such as "telling the time"), it's also called a "dedicated computer". Most hidden computers are "almost free" ($1-$50).

A *pocket computer* fits in your pocket. It looks like a pocket calculator, but includes keys you can press for typing all the letters of the alphabet, so that you can store names and addresses and communicate with the computer by using English words. Most pocket computers cost between $50 and $100.

To let the pocket computer fit in your pocket, the keys on its keyboard are very tiny and are therefore hard to press (unless you have tiny fingers). Most adults that have big fingers prefer a *lap computer* instead, which looks like a pocket computer but has larger keys and a larger screen. It's too large to fit in your pocket, but it fits nicely in your lap. Since it's about the size of a notebook, it's also called a "notebook computer"; and since it's truly easy to carry under your arm, it's called a "truly portable computer". It's also called a "pocket computer for elephants", since it's the kind of pocket computer an elephant would have, if an elephant had pockets. Most of those lap computers cost between $500 and $3000.

A *TV computer* looks like a lap computer but lacks a screen; instead of including a screen, it attaches to your home's TV, so that whatever you type on the keyboard appears on TV! It turns you into a TV star! The TV screen shows everything you typed, and also shows the computer's answers to your questions. Most TV computers cost between $50 and $1000.

Unfortunately, the picture on a TV screen is slightly fuzzy. To get a sharper picture from a computer, attach the computer to a *computer monitor*, which is a super-high-quality TV especially designed for being attached to a computer. A symbol (such as a letter of the alphabet or a digit or a punctuation mark) is called a *character*; a computer monitor is sharp enough so that its screen can clearly display about 2000 characters simultaneously. The 2000 characters are typically arranged in 25 rows, each containing 80 characters. A computer attached to a monitor and displaying about 2000 characters simultaneously is called a *full-monitor computer*. Most full-monitor computers cost between $600 and $6000.

A person who uses a computer is called a *user*. The typical microcomputer handles just one user at a time. If the microcomputer is very large—or if you have a minicomputer or maxicomputer instead—you can attach several keyboards and screens to it simultaneously, so that the computer can handle many users at once. That's called a *multi-user computer* (or *multi-user system*). The cheapest decent multi-user computer is the *Tandy 6000*, which costs about $5000. Other multi-user computers are much more expensive. Most minicomputers and maxicomputers are multi-user. Yes, multi-user computers can cost up to $10,000,000!

FOUR HOUSES

THE HOUSE THAT JACK BUILT

Commodore was started by Jack Tramiel.

Jack began his career by being in the wrong place at the wrong time: he was a Jew in Poland during World War 2. He was thrown into the Auschwitz concentration camp, where he learned to view life as a war to survive. When he finally escaped from the camp, he moved to the United States and started an aggressive, ruthless business whose motto for survival was, "Business is war!"

He called the business "Commodore". At first, it was a company that just repaired typewriters; but the company grew rapidly. Then Commodore built its own electronic calculators and sold them. Finally, Commodore made a bolder move, and decided to build entire computers.

To get a design for a computer, Commodore tried to buy Apple Computer Company, and almost succeeded; but Apple wanted $15,000 more than Commodore would agree to, so the deal never came off. Commodore has regretted it ever since.

Commodore offered to build computers for Radio Shack, which wasn't selling any computers yet. Radio Shack said, "Build us a prototype, and we'll think about it." Commodore built a prototype and showed it to Radio Shack. Radio Shack said, "Thanks, but your idea of selling low-cost computers was so convincing that we decided to build our own. Thanks for the idea." That's how Radio Shack began selling computers!

Having been rebuffed by both Apple and Radio Shack, Jack ("Business is War") Tramiel decided to fight back, by building a computer that was better and cheaper than anything Apple and Radio Shack had to offer. Commodore called its new computer the *Pet*, because Commodore's marketing director was the same guy who invented the Pet Rock. He figured that if people were stupid enough to buy a Pet Rock, they'd *really* love a Pet computer!

The Pet computer was a smashing success, and Commodore became the top-selling computer company. But that success didn't last long: Commodore made several blunders and slipped quickly into third place, behind Radio Shack (which became the top seller) and Apple (which became number 2). Commodore came out with a souped-up Pet, called the *Commodore Business Machine (CBM)*, but even that souped-up version wasn't enough to let Commodore rise above the number 3 spot.

Because of World War 2, Jack was scared of Nazis—and also of the Japanese. He became extremely frightened that the Japanese would suddenly invade the United States, by flooding America with low-cost Japanese computers that would put Commodore and other American companies out of business. So he stood up in front of a meeting of his engineers and gave a paranoid speech in which he screamed, "The Japanese are coming! The Japanese are coming! So *we* will become the Japanese!" Then he laid out his bold plan: Commodore would produce a computer that would out-Japanese the Japanese. It would be even cheaper and better than anything the Japanese could build. Specifically, Commodore would build the first computer in the whole world that would hook up to a color TV, display colors on the TV, produce three-part harmony through the TV's speaker, and retail for under $300.

That was a bold move. Commodore's engineers said, "It's impossible to build a color computer that cheaply!" Jack just said, "Do it!" Commodore's engineers finally managed to do it.

Because the new computer was feminine and foxy, Commodore wanted to call it the "Vixen". But Commodore discovered that a "Vixen" computer couldn't sell in Germany, because "Vixen" sounds like the German word "Wichsen", which is obscene.

Commodore hastily changed the name to *Vic*, and ran ads in magazines and on TV for the "Vic" computer. But that got Commodore into even worse trouble, because "Vic" sounds like the German word "Ficke", which is even *more* obscene!

Although Commodore still calls it the "Vic" in the United States, Commodore calls it the "VC" computer in Germany, and pretends that "VC" stands for "Volks Computer".

In spite of Commodore's troubles with German, Commodore's computer quickly became very popular in the United States, Germany, England, and even Japan. Commodore quickly sold over a million of them! Commodore's Vic became the best-selling computer in the whole world!

A general rule of thumb in the computer industry is: prices tend to drop about 30% each year. So although Commodore originally sold the Vic for slightly under $300 ($299.95), today you can buy it for about $50 at most toy stores and department stores.

After inventing the Vic, Commodore invented a fancier version, called the *Commodore 64*. It originally listed for slightly under $600 ($599.95), but today you can buy it for under $150 at most toy stores and department stores. Like the Vic, the Commodore 64 became extremely popular, and Commodore sold over a million of them. Because the Commodore 64 was so inexpensive and yet so powerful, many programmers who had written programs for Apple computers rewrote their programs so that they'd also work on the Commodore 64. Today, the Commodore 64 does almost everything the Apple can, and yet costs much less.

After the Commodore 64 became successful, Jack Tramiel wanted to hire his sons to help run Commodore; but the other heads of Commodore, who were major stockholders, refused to deal with Jack's sons. So Jack quit. He also sold his 2 million shares of Commodore stock, at $40 per share, netting himself 80 million dollars in cash. He used some of that cash to take his wife on a trip around the world. But when he and his wife reached Japan, the heads of Japanese computer companies told him, "Jack, we're so glad you quit Commodore, because now we can enter the American computer marketplace without having to fight you."

That comment scared Jack. He didn't want to let the Japanese invade the U.S. computer market. So he started a second computer company, called "Tramiel Associates", whose sole purpose was to stop the Japanese invasion.

Tramiel Associates bought Atari (a California company that had been a division of Warner) and successfully stopped the Japanese invasion by lowering the retail price of Atari computers to $75—a price so low that the Japanese can't compete against it. Today, Jack and his sons run Atari. He competes against the company he founded: Commodore.

He also competes against Apple, by selling his own low-cost imitation of Apple's Macintosh computer. Jack's low-cost imitation is nicknamed the "Jackintosh". It's also called the *Atari 520 ST*. It sells for $799, which is less than half the price of a Macintosh.

Jack's the only exciting guy in the whole computer industry; he's the only guy preventing the industry from stagnating. I hope his plans succeed. Since Jack is ruthless ("Business is War"), he's an effective manager but also a bluffer: we'll have to wait to see whether his Jackintosh and other grand schemes become everything he promises.

After Jack left Commodore, Commodore invented two more computers, called the *Commodore 16* and the *Commodore Plus 4*; but they had serious flaws and didn't sell well. Then Commodore began selling two improved computers, called the *Commodore 128* and the *Amiga*, which are wonderful.

(17) Computer industry: four houses

THE HOUSE THAT STEVE BUILT

Apple Computer Company was started by two kids named "Steve". When they were in high school, they started an illegal business: they built and sold *blue boxes*, which were boxes that you attached to your phone so that you could illegally make long distance calls free. They sold 200 blue boxes at $880 each, giving them a total of $16,000 in illegal money.

After that immoral beginning, they finally did something right: they invented, built, and sold Apple computers. Steve Wozniak designed the circuitry; Steve Jobs did the marketing.

When Steve Wozniak was designing the Apple circuitry, he was also employed by Hewlett-Packard, so he offered all his designs to Hewlett-Packard. But Hewlett-Packard said it wasn't interested. Hewlett-Packard has regretted that decision ever since.

The original Apple computer looked pathetic. But then Steve & Steve invented a slicker version, called the *Apple 2*. It became a smashing success, because it was the first low-cost computer (under $1000) that could display colors on a TV. The Apple 2 remained the *only* computer for under $1000 that could display colors on a TV, until many years later, when Commodore invented the Vic, which was even cheaper (under $300).

At first, people used the Apple 2 to play games, and didn't take the computer seriously. But then two important events occurred, that changed the world's feelings about Apple.

First, the Minnesota state government decided to buy lots of Apple 2 computers, put them in Minnesota schools, and write programs for them. That state agency, called the *Minnesota Educational Computing Consortium (MECC)*, then distributed the programs free to other schools across America. Soon, schools across America discovered that personal computers could be useful in education, and that the only programs available came from Minnesota and worked only on Apple computers. To use those programs, schools across America started buying Apples. Then *those* schools wrote additional programs for the Apples they had bought. Apple became the "standard" computer for education—just because of the chain reaction that started with a chance event in Minnesota.

Next, a graduate student at the Harvard Business School and his friend at M.I.T. got together and wrote an accounting program called *Visicalc*. They wrote it for the Apple 2 computer, because it was the only low-cost computer that could store long programs reliably. (Commodore's computers couldn't handle long programs, and Radio Shack's computers were unreliable until the following year.) The Visicalc accounting program was so wonderful that accountants and business managers all over the country bought it—and therefore had to buy Apple computers to run it on. Visicalc was better than any accounting program that had been invented on even the largest IBM maxicomputers. Visicalc proved that little Apples could be more convenient than even the most gigantic IBM. Accountants fell so much in love with Visicalc that when Olivia Newton John began singing her famous song, "Let's Get Physical", accountants thought she was singing, "Let's Get Visicalc" and began dancing to work.

Later, Visicalc became available for other computers; but at first, Visicalc required an Apple, and Visicalc's success led to the success of Apple. In a typical large corporation, the corporate accountant wanted to buy an Apple with Visicalc, but the head of the corporation's data-processing department refused to spend any money on those "junky personal computers", so the accountant who wanted Visicalc had to resort to an old business trick: he lied. He pretended that he was spending the $2000 for "typewriters" but bought an Apple instead. He sneaked it into the company when nobody was looking, and put it on his desk. That happened all across America, so that all of America's large corporations soon had thousands of Apples sitting on the desks of the corporate accountants and managers, all disguised as "typewriters" or "word processors". Yes, Apple computers infiltrated American corporations by subversion. It was an underground movement that annoyed IBM so much that IBM eventually decided to invent a personal computer of its own.

Meanwhile, Apple Computer Corporation invented a slightly improved Apple 2, called the *Apple 2 plus*; then a further little improvement, called the *Apple 2e* (the "e" stands for "enhanced"); and then a further improvement, which also weighed less and was easier to carry and called the *Apple 2c* (the "c" stands for "compact and cute"). Those computers are all very similar to each other, so that most programs written for the Apple 2 also work on the Apple 2 plus, the Apple 2e, and the Apple 2c.

Today, Apple Computer Incorporated makes the Apple 2e and Apple 2c. The earlier models (the 2 and 2 plus) are obsolete.

The Apple 2e and Apple 2c each cost about $800. (The exact price depends on which options you choose.)

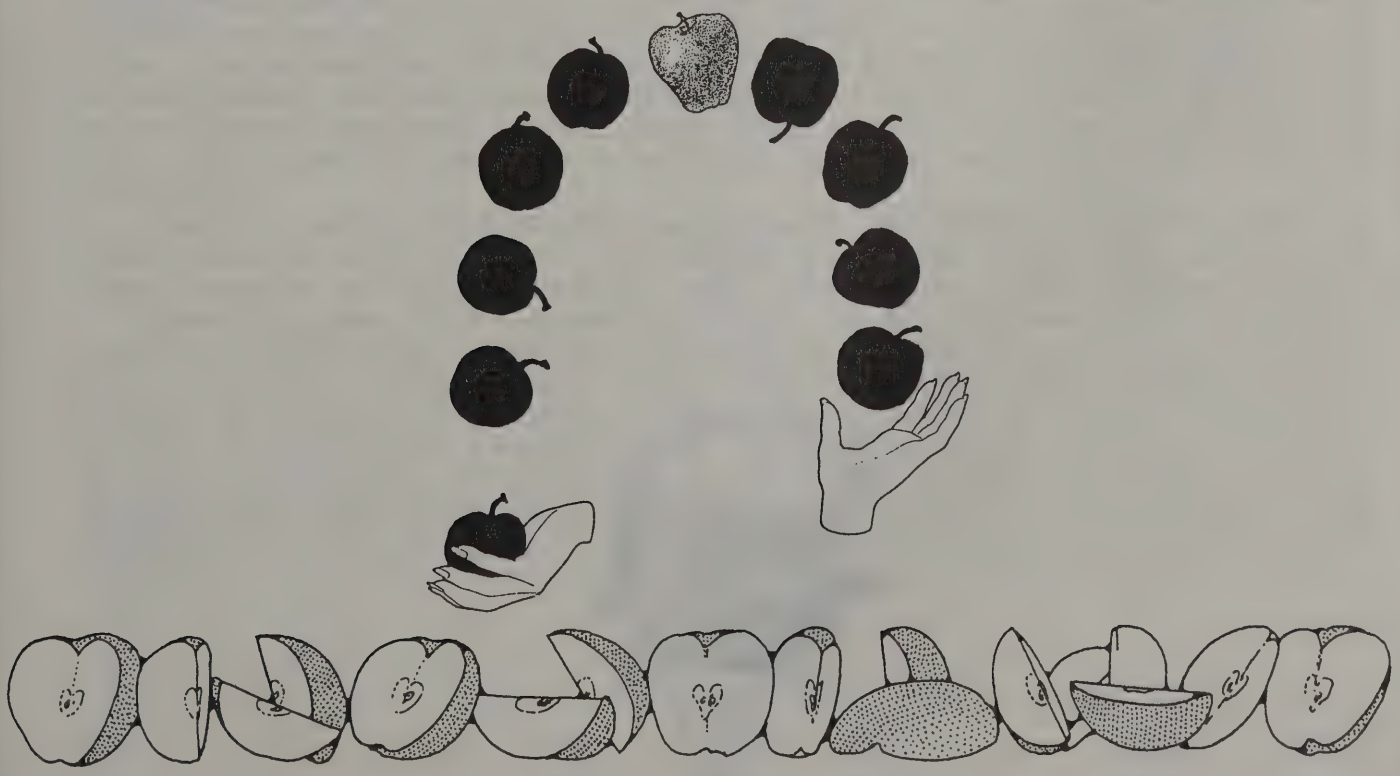
Apple Computer Incorporated has also tried to make fancier computers that cost more, for businesses that can afford to pay for luxury. Apple's first luxurious computer was the *Apple 3*. Although it sported many fancy features, it didn't sell well, because it was too expensive, and because it was designed too hastily to be reliable. When IBM finally invented the wonderful IBM Personal Computer, the Apple 3 looked overpriced and silly by comparison, and Apple stopped making it.

Apple's next attempt at a luxurious computer was the *Lisa*, which was named after Steve Jobs' out-of-wedlock daughter. Although it had beautiful graphics and was remarkably easy to learn how to use, it was too expensive to be practical, and so Apple has stopped making it.

Apple's third attempt at a luxurious computer was the *Macintosh*, which was a stripped-down version of the Lisa. The Macintosh's price was low enough to make it popular.

The Macintosh is easy to learn how to use, but when you try using it for business applications you spend lots of time waiting, because the Macintosh is slow. Its slow speed, stripped-down keyboard, and not-quite-low-enough price make it less practical for business applications than IBM's computers. But since Apple is continually improving the Macintosh and lowering the price, the Macintosh may someday become even more popular than the IBM PC.

Apple's founders—Steve & Steve—have quit Apple and started new companies. I wish them well. Now Apple is headed by John Sculley, who used to be a vice-president of Pepsi. He made Pepsi the #2 soft drink, just behind Coke; now he's keeping Apple the #2 microcomputer company, just behind IBM.



TRASH FROM TEXAS

Howdy, pardner! Fort Worth, Texas, is the headquarters of Tandy, which makes a greater variety of microcomputers than any other manufacturer. Tandy makes thirteen different microcomputers: the *Tandy 200*, *Tandy 600*, *Tandy 1000*, *Tandy 1200*, *Tandy 2000*, *Tandy 3000*, *Tandy 6000*, *Pocket Computer 3A*, *Pocket Computer 4*, *Pocket Computer 5*, *Model 4D*, *Model 100*, and *Color Computer 2*.

Tandy's computers are often called "TRS" computers; the "TRS" stands for "Tandy's Radio Shack". People who dislike TRS computers add the letters A and H, and call them "TRASH" computers.

Folks who use Tandy's computers are therefore called "trash collectors". On the other hand, Apple lovers are called "worms" (or "pie people" or "fruits" or "suffering from Appleplexy"); Commodore lovers are called "boat people" (or "swabbies" or "deck ducks" or "wetbacks"); IBM lovers are called "blue bloods" (because old IBM computers were blue); and kids who play with Atari computers are called "Atari-eyed dreamers & screamers".

Today, Tandy's computers are built more reliably than the computers of any of its competitors. Moreover, Tandy charges less for repairs than any of its competitors.

Prices for Tandy computers are amazingly low. Tandy's imitations of IBM's computers work just as well as IBM's and yet cost hundreds of dollars less.

Tandy's main problem is its advertising: its advertisements are too straight-laced and boring; they lack the flair and excitement of the ads created by IBM, Apple, and Commodore. When most people see a Tandy ad, they fall asleep.

But although Tandy's ads are boring, at least they're honest. Most of the ads created by Apple and Commodore are dishonest and misleading.

Practically all the computer equipment that Tandy sells is "grade B+": it's adequate and a solid value. When you buy a new product from Apple or Commodore, you're taking more of a risk: the product might be terrific, but it might also be terrible; you won't know until you try it. Tandy is the safest company to buy from. Even IBM isn't as safe as Tandy: thousands of customers have complained that IBM's newest computer (the IBM PC AT) is unreliable, though IBM has recently fixed the problem.

Also, Tandy is the only popular microcomputer maker whose headquarters provides a reliable hot-line for answering your technical questions.

IBM bases its entire marketing strategy on one word: *react*. IBM never creates a new kind of computer; instead, IBM watches its competitors' products, notices which ones sell well, and then designs a product that meets the same needs better.

Even IBM's own name is a reaction. IBM was started by Tom Watson. He'd been a salesman for National Cash Register (NCR), but got fired, so he took over a competing company (CTR) and vowed to make it even bigger than National Cash Register. To be bigger than "National", he called his company "International"; to be bigger than a "Cash Register" company, he bragged that his company would sell *all* kinds of "Business Machines". That's how the name "International Business Machine Corp."—IBM—was hatched. IBM quickly outgrew NCR.

IBM sold lots of business machines, especially to the U.S. Census Bureau. But in 1951, Remington Rand Corp. (which later merged with Sperry) developed the Univac computer and convinced the Census to use it instead of IBM's non-computerized equipment. To react, IBM quickly invented its own computers, which were more practical than the Univac. IBM quickly became the #1 computer company—and Sperry's Univac dropped to #2.

IBM's computers were large. IBM ignored the whole concept of microcomputers. But when lots of IBM's customers began buying Apple 2 microcomputers to do Visicalc spreadsheets, IBM reacted, by developing the *IBM Personal Computer (IBM PC)*, which did everything that Apple 2 computers could do, but better. IBM developed its PC secretly, and didn't reveal the secret until August 12, 1981. The PC was a smashing success: IBM quickly became the #1 microcomputer company—and Apple dropped to #2.

Today, the IBM PC is the best-selling microcomputer for business. A greater variety of high-quality business programs is available for the IBM PC than for any other microcomputer. It's become the standard against which all other microcomputers are compared. It's wonderful; and even though some of IBM's competitors might be slightly cheaper or slightly fancier, you can't do much better than buy an IBM PC. The best business programs are designed for the IBM PC; so to use the best business programs, you must buy an IBM PC (or a clone).

How IBM invented the PC. Although certain subgroups at IBM had previously invented other microcomputers (the IBM 5100 and the IBM System 23 Datamaster), nobody took them seriously—not even IBM. The IBM PC, on the other hand, was a *very* serious project. To invent the IBM PC, IBM held a secret contest (known only to IBM's management): IBM created three research teams who competed against each other. The winner was the research team headed by Philip "Don" Estridge in Boca Raton, Florida. His team examined everything that had been created by all the other microcomputer companies (Apple, Radio Shack, Commodore, etc.) and combined their best ideas, to produce a relatively low-cost computer better than all competitors.

The only mistake Don made was the keyboard. Don had his own personal theory about where to put the keys on the keyboard and how to label them. His theory, though cleverly original, was wrong: most typists prefer a more traditional layout.

Newer computers. IBM's newest computers—the *PC Junior* and the *PC AT*—sport keyboards that are more traditional, though still far from ideal. Unfortunately, the Junior's internal circuits differ enough from the original PC so that the Junior refuses to run some of the original PC programs; and the AT, though super-fast, can also be unreliable. Those problems made the machines unpopular, so Don's bosses decided to stop manufacturing the PC Junior, changed the way the PC AT is manufactured, kicked Don out of Boca Raton, and hid him in an obscure part of the company. A few months later, when he flew on a Delta jet, the jet crashed and killed him.

WOW! YOU FINISHED CHAPTER 1: THE COMPUTER INDUSTRY! TURN THE PAGE FOR CHAPTER 2: THE COMPUTER'S PARTS.

SCREENS

WHAT THE SCREEN SHOWS

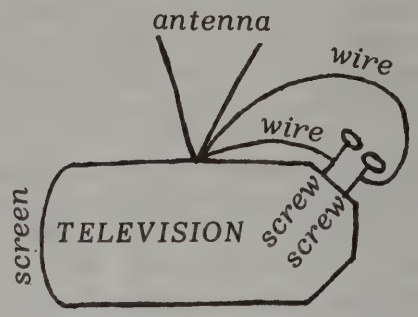
The computer comes with a *keyboard*, which resembles a typewriter's. To give the computer a command, put your fingers on the keyboard, and start typing the command. As you type the command, it appears on the computer's *screen*, which is an ordinary TV or something that resembles a TV.

When you finish typing the command, the screen shows the computer's reply. For example, if you command the computer to solve a problem, the screen shows the problem you typed, and also shows the computer's solution.

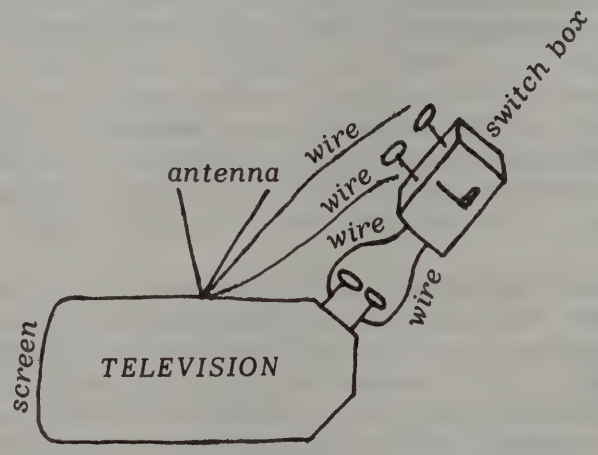
TELEVISIONS

If your computer can be attached to an ordinary TV, it's called a *TV computer*. Here's how to attach a TV computer to a TV.

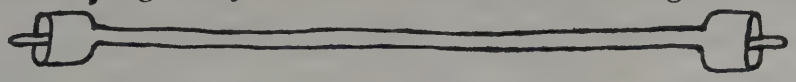
Look at your TV's antenna. Wires run from the antenna to two screws, which are on the back of the TV:



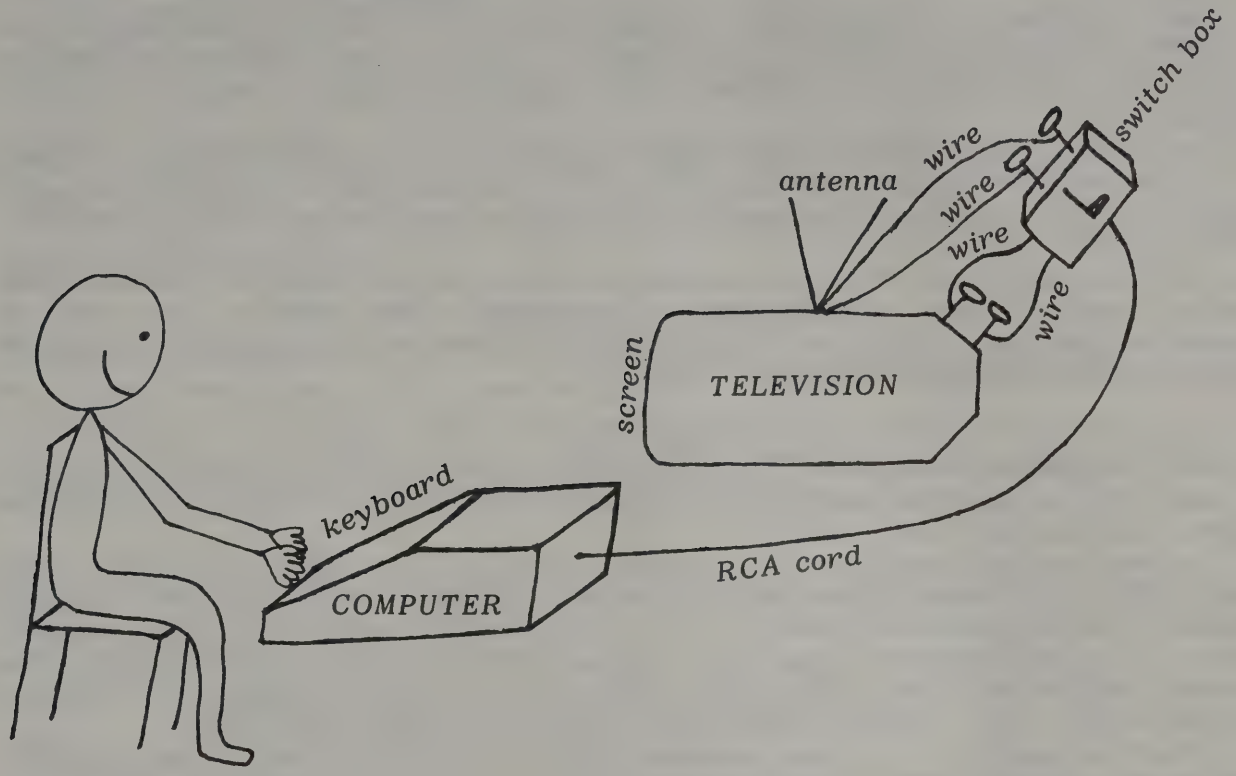
Loosen the two screws, to release the antenna. When you buy a TV computer, the salesperson also gives you a *switch box*. Attach that box to the two screws you loosened:



The salesperson also gives you an electrical cord looking like this:



It's called an *RCA cord*, because RCA invented it. Plug one end of that cord into your computer, and plug the opposite end into the switch box. Attach the antenna's wires to the screws on the switch box:



The switch box has a switch on it. If you move the switch towards the antenna, you have a normal TV, so you can watch Johnny Carson. If you move the switch towards the computer's RCA cord, your TV's controlled by the computer, so that the computer can write messages on your TV screen.

So by moving the switch, you can make your TV act either normally or computerized. Your family will argue about which way to move the switch.

That switch box is the same kind used by video games (such as Mattel Intellivision, Atari VCS, and Colecovision). When you buy a TV computer, the salesperson will give you the switch box and RCA cord, free!

To use the computer, you must turn your TV's dial to the correct channel. Most computers use channels 3 and 4.

To get a sharp picture on your TV screen, *avoid* the channel used by your local TV station. For example, if you live in Boston or New York City, NBC hogs channel 4, so *avoid* channel 4; put your computer on channel 3 instead. (Turn your TV's dial to channel 3, and also turn the computer's switch to channel 3.)

Although most computers (such as Commodore and Radio Shack) use channels 3 and 4, some computers (such as Atari) use channels 2 and 3 instead, and some other computers use channels 10 and 33 and 34 instead.

If the image on your TV screen looks fuzzy—so that you can barely read the computer's writing—adjust the TV's "fine tuning" knob.

What the TV computer can do. Besides writing messages on your TV's screen, the computer can also draw its own *pictures* on the TV! And if your TV has color, you'll see the pictures in color!

When you watch Johnny Carson on TV, the size of Johnny's face depends on the size of your TV's screen. If your TV's screen is tiny (less than 12 inches), Johnny's face looks small; if your TV's screen is 25 inches, Johnny's face looks larger; and if you have a projection TV with a gigantic 60-inch screen, Johnny's face looks gigantic. The same holds true for the messages and pictures sent to the TV by the computer: the larger the TV's screen, the more magnified the computer's messages and pictures.

How much does the TV display? The computer makes the TV's screen show lots of words, numbers, and formulas. Those words, numbers, and formulas are made of *characters*: each character is a letter of the alphabet, a digit, or any other symbol you can type.

The perfect TV computer makes the TV display 25 lines of information, and each line of information contains 40 characters. So by using a perfect TV computer, the total number of characters you see on the TV screen simultaneously is "25 times 40", which is 1000.

Unfortunately, most TV computers sold today are slightly less than perfect: they display slightly fewer than 25 lines of information and slightly fewer than 40 characters per line, and so the total number of characters you see on the TV screen simultaneously is slightly less than 1000.

MONITORS

A *computer monitor* resembles a TV, but produces a sharper picture and costs more.

Like a TV, a computer monitor uses a picture tube. The tube in a TV or monitor is called a *cathode-ray tube (CRT)*.

The monitor can be either *stand-alone* or *built-in*. A *stand-alone monitor* looks like a TV, but doesn't have an antenna and doesn't have a dial for selecting channels: the only channel you can get is "computer". It doesn't need a switch box: instead, the computer's RCA cord (or a similar cord) plugs directly into a hole in the monitor. Before you buy a computer that uses a stand-alone monitor, ask whether the computer's price includes the monitor: the monitor might cost extra.

A *built-in monitor* is a screen permanently attached to the rest of the computer: the unit that contains the computer's main circuits also contains the monitor.

Colors. When you buy a TV, you ask for either "color" or "black-and-white". Similarly, when you buy a computer monitor, you ask for either *color* or *monochrome*. A color monitor displays all the colors of the rainbow; a monochrome monitor displays just dark-and-light.

The cheapest monochrome screen displays black-and-white, but fancier monochrome screens look tinted. For example, you can buy a *green-screen monitor*, which displays dark green instead of black, and displays light green instead of white. You can also buy an *amber-screen monitor*, which displays dark orange instead of black, and displays light orange instead of white. Green-screen and amber-screen monitors are easier on your eyes than black-and-white monitors, because black-and-white monitors tend to produce too much glare.

A color TV costs more than black-and-white. Similarly, a color monitor (that displays all the colors of the rainbow) costs more than a monochrome monitor.

Most monochrome monitors cost between $100 and $200. Most color monitors cost between $200 and $500.

Sizes. The standard size for a monitor's screen is 12 inches (measured diagonally). Some monochrome monitors use a slightly smaller screen (9 inches), so that the monitor is smaller and weighs less and is slightly easier to carry.

A computer that includes a built-in 9-inch monitor weighs about 20 pounds altogether. Such a computer system is light enough and small enough so that it fits in a bag or case that you can carry in one hand—if you have a strong arm. Such a computer is advertised as being *transportable*, though people who have weak arms feel the word *luggable* is more accurate.

Resolution. If the cable that runs from the monitor to the computer is extra-fat (so that it contains many wires), the monitor produces an extra-sharp image.

A color monitor with an extra-fat cable is called a *red-green-blue monitor (RGB monitor)*; the cable contains one wire to transmit red, a second wire to transmit green, a third wire to transmit blue, plus additional wires for further enhancements.

A monochrome monitor with an extra-fat cable is called a *transistor-transistor-logic monitor (TTL monitor)*; the cable contains one wire to transmit the fundamental picture, plus additional wires for further enhancements.

If your monitor's cable is *not* fat, the picture isn't as sharp. If your monitor's cable is just a thin RCA cord, your monitor's called a *composite monitor*; its picture is worse than a fat-cable monitor, though not as bad as a plain TV.

Some computers can send the monitor a signal that makes the monitor display 80 characters per line. To fit that many characters on a line, the characters are made tiny. To display those tiny characters on the screen clearly, the monitor must either have a fat cable (to produce a sharp picture) or be monochrome; it must *not* be a composite color monitor. (If you try to display tiny characters on a composite color monitor, the fuzzy colors bleed into each other, and so the characters on the screen are hard to read.)

The typical monitor displays 25 80-character lines, so that you see 2000 characters simultaneously.

VIDEO TERMINALS

A *video-display terminal (VDT)* is a monitor that has a keyboard attached to it, and which can communicate with a large computer.

If 200 people are using a maxicomputer simultaneously, only one of them is sitting at the maxicomputer's main console. The other 199 people typically sit at 199 VDT's, which are in different rooms or even different cities.

LIQUID CRYSTALS

If your computer is tiny, it comes with a tiny screen, called a *liquid-crystal display (LCD)*. That's the kind of screen you see on digital watches, pocket calculators, pocket computers, and lap computers.

Because an LCD screen is so small and uses so little electricity, it can run on batteries. A traditional picture tube can*not* run on batteries. So if your computer system runs on batteries, its screen is an LCD.

An LCD screen displays black characters on a white background. The screen consists of thousands of tiny crystals. Each crystal is normally white, but temporarily changes to black when an electrical charge passes through it.

A traditional picture tube emits light; that's why you can watch TV even in a dark room. But an LCD screen does *not* emit light; you can*not* read an LCD screen in a dark room. You must turn on a light in the room, to see which of the crystals are white and which are black.

Some LCD screens include a light you can turn on, to see the screen more easily. But since that light consumes a lot of electricity, you'll wear out the battery if you keep the light on too long.

Some LCD screens include a light that stays on continuously. Those LCD screens can't run on batteries; you must plug them into a wall socket instead.

Although the crystals can change from white to black, they appear black only if you look at them from the correct angle, and if the light in the room is positioned correctly. If you move your head or the light, the black crystals will appear very light gray instead of black, and you'll have trouble reading the message they're trying to display. So if you have trouble seeing the message on an LCD screen, move your head or the light or the screen, until the message darkens.

LCD screens have several advantages over traditional picture tubes: LCD screens consume less electricity, weigh less, and are less bulky. On the other hand, most LCD screens don't work in dark rooms, can't display colors, exhibit poor contrast (unless your eye's at exactly the right angle from the screen), respond too slowly to computer commands, and suffer from poor resolution: either they don't display enough characters per line, or they don't display enough lines on the screen, or the graphics are too crude, or the characters are poorly shaped. Because of those disadvantages, I recommend instead a traditional picture tube (TV or monitor or video terminal), unless you absolutely need the screen to be small and battery-operated.

CHIPS

MAIN CIRCUITRY

The computer's *main circuitry* remembers your problem and calculates the solution.

The main circuitry is hidden. If your computer is very small (such as a pocket computer or a lap computer or a small TV computer), the main circuitry hides inside the keyboard. If your computer's main circuitry is too large to fit in a keyboard, the main circuitry hides in a separate box, called the *system unit*.

PC boards. If you pry open the keyboard or system unit, to see the main circuitry inside, you'll see a green plastic board. An electrical wiring diagram's printed on the board. But since the diagram's printed in copper (instead of in ink), the diagram conducts electricity; so it isn't just a diagram of an electrical circuit: it *is* an electrical circuit!

The green plastic board—including the circuit printed on it—is called a *printed-circuit board (PC board)*. Each wire that's stamped onto the PC board is called a *trace*.

The typical computer contains *several* PC boards. In your computer, the largest and most important PC board is called the *motherboard*. It lies flat on the bottom of the system unit (or keyboard). The other PC boards are smaller; those little baby boards (about the size of a postcard) are called *PC cards*.

Caterpillars. On each PC board, you'll see black bumps. If you look closely at a black bump, you'll see it has tiny little legs, so that it looks like a black caterpillar. The caterpillars come in various sizes: in a typical computer, the shortest caterpillars are three-quarters of an inch long and have 7 pairs of legs; the longest are two inches long and have 20 pairs of legs.

Although each black caterpillar has legs, it doesn't move. It's permanently mounted on the PC board.

Each leg is made of tin and is called a *pin*.

Sadistic hobbyists play a game where they yank the caterpillars from the PC board and then throw the caterpillars across the room. That game's called "tin-pin bowling".

Hiding the chips. Hidden inside the caterpillar is a metal square, called a *chip*, which is very tiny: it's just an eighth of an inch long, an eighth of an inch wide, and a hundredth of an inch thick! On that tiny metal chip are etched *thousands* of microscopic electronic circuits! Because all those circuits are on the chip, the chip's called an *integrated circuit (IC)*.

How chips are designed. To design the circuits inside the chip, the manufacturer hires an artist, who draws on paper a big sketch of the circuits. It helps if the artist also has a degree in engineering—and knows how to use another computer to help draw all the lines.

After the big sketch is drawn, it is photographed.

Have you ever photographed your friend and asked the photography store for an "enlargement"? To produce a chip, the chip's manufacturer does just the opposite: it photographs the sketch but produces a "reduction" that's just an eighth of an inch on each side! And whereas a photograph of your friend is made on treated paper, the tiny photograph of the chip's circuitry consists of metal and semiconductors on treated silicon, so that the photograph is an actual working circuit! That photographic process is called *photolithography* (or *photolith*).

Many copies of that photograph are made on a large silicon wafer. Then a cookie cutter slices the wafer into hundreds of chips. Each chip is put into its own caterpillar.

The caterpillar's only purpose is to hide and protect the chip inside it; the caterpillar's just a strange-looking package that contains the chip. Because the caterpillar's a package that has two rows of legs, it's called a *dual in-line package (DIP)*. That DIP's only purpose is to hide the chip.

Computer hobbyists are always talking about chips and DIPs. That's why, at parties, computer hobbyists serve chips and dips. And that's why computer hobbyists are called "dipchips".

Buying chips. If you enter a computer store and ask to buy a chip, the store will also give you the chip's DIP (the entire caterpillar). Since you've asked for a chip but also received a DIP, you might get confused and think that the caterpillar (the DIP) is the chip. But that caterpillar is *not* the chip; the chip hides inside the caterpillar.

The typical caterpillar-and-chip costs $3. You might pay somewhat more or somewhat less, depending on how fancy the chip's circuitry is.

If the circuits in a chip are defective, it's called a "buffalo chip". Folks who dislike that tacky term say "potato chip" or "chocolate chip" instead, like this: "Hey, the computer's not working! It must be made of chocolate chips!"

How chips chat. The chip inside the caterpillar acts as the caterpillar's brain. The caterpillar also contains a "nervous system", made of thin wires that run from the brain (the chip) to the legs (the pins). The wires in the caterpillar's nervous system are very thin: each wire's diameter is about half of a thousandth of an inch.

It's fun to give each caterpillar a nickname. For example, suppose your computer contains a caterpillar named Carl and another caterpillar named Katy. When Carl wants to send electrical signals to Katy, the signals go from his brain (the chip) through his nervous system to his legs (pins). Each of his pins is attached to a trace (wire) on the PC board. The electrical signals travel from his pins to the PC board's traces, which carry the signals across the PC board, until the signals reach Katy's pins. Then the signals travel through her nervous system, until they reach her brain (the chip).

Chip technologies. Chips can be manufactured in two ways. The old way's called *bipolar*. The new way's called *metal-oxide semiconductor*, which is abbreviated *MOS*, which is pronounced "moss". The new way (MOS) is more popular, because it costs less, consumes less electricity, and can hold more circuitry inside the chip.

Microcomputers use only MOS. Minicomputers and maxicomputers use mainly MOS chips but also contain a few bipolar chips, because bipolar chips have one (and only one) advantage over MOS chips: bipolar chips work faster.

The most popular kind of MOS is called *n-channel MOS (NMOS)*. An alternative, called *complementary MOS (CMOS)*, consumes even less electricity but can't hold as much circuitry inside the chip. CMOS chips are used in simple-minded battery-operated computers, such as digital watches, pocket calculators, pocket computers, and lap computers.

Four purposes. Each chip serves a purpose. If the chip's purpose is to "think", it's called a *processor chip*. If the chip's purpose is to "remember" information, it's called a *memory chip*. If the chip's purpose is to help devices communicate with each other, it's called an *interface chip*. If the chip's purpose is to act as a slave and helper to other chips, it's called a *support chip*.

So a chip is either a processor chip or a memory chip or an interface chip or a support chip—or it's a combination chip that accomplishes *several* purposes.

CPU

The part of the computer that thinks ("the brain") is called the *processor* (or *central processing unit* or *CPU*).

In a maxicomputer or minicomputer, the processor consists of several chips, which are *processor chips*.

In a microcomputer, the processor is so small that it consists of just a single chip, called a *microprocessor*. It sits on the motherboard. Yes, in a typical microcomputer, the part that does all the thinking is just a tiny square of metal (less than ¼" on each side) that costs under $10!

Popular microprocessors. The most popular microprocessors are manufactured by four companies. Let's look at them.

Zilog, which is owned by Exxon, makes the *Z-80A microprocessor*, which is super-cheap: it costs just $2! It's in the Radio Shack model 4 & 4P & 4D, the Radio Shack model 12, the Coleco Adam, the Epson Geneva, and the Kaypro 2 & 2X.

MOS Technology, which is part of Commodore, makes the *6502 microprocessor*, which costs $3. It's in the Atari 800 & 800XL, the Apple 2 & 2+, and the original Apple 2e. The Apple 2c and the newest version of the Apple 2e use a souped-up version called the *65C02*, which is made of CMOS and costs $13. The Commodore Vic & 64 & 128 use a souped-up version called the *6510*, which costs $20.

Motorola makes the *6809E microprocessor*, which is cute, costs just $6, and is in the Radio Shack Color Computer 2. Motorola also makes the *68000 microprocessor*, which is expensive ($10); it's in the Apple Macintosh, the Radio Shack model 6000, the Commodore Amiga, and the Atari 520 ST.

Intel, which is partly owned by IBM, makes the *8088 microprocessor*, which costs $7. It's become the standard for serious business computers: it's in the IBM PC, the IBM PC Junior, the IBM Portable PC, the IBM PC XT, and most computers that imitate them. Intel also makes the *80286 microprocessor*, which is a souped-up 8088, runs faster than the 8088, and costs $130; it's in the IBM PC AT and computers that imitate it.

Each of those microprocessors comes in a DIP (caterpillar) that has 40 pins (20 pairs of pins)—except for the Motorola 68000 and the Intel 80286, which are super-fancy and have 64 pins instead.

What do you really pay? Those prices are what mail-order electronics supply houses charge for "the plain version, quantity one". If you look inside your computer, you'll probably find a slightly fancier version of the microprocessor instead (which costs more), but on the other hand your computer's manufacturer got a quantity discount.

The price of a microcomputer always includes a microprocessor at no extra charge. You pay for the microprocessor separately only if you're inventing your own computer, or if you buy a computer that breaks and needs new parts.

Notice that a microprocessor is cheap: the quantity-one price is between $2 and $130, depending on which microprocessor you get. Yes, the computer's "brain" costs between $2 and $130, and is just a tiny caterpillar containing a tiny square of metal that's about an eighth of an inch on each side!

But the computer that contains the microprocessor is much more expensive: the entire computer can cost *thousands* of dollars. That's because the microprocessor is just a tiny part of the computer. In addition to the microprocessor, you need memory chips, interface chips, and support chips; you also need PC boards to put the chips on; you also need a keyboard, a screen, and other I/O devices; and you need lots of other goodies too!

So even though the microprocessor can be worth as little as $2, when you ask a salesperson "How much does the whole computer cost", the answer will probably be "several thousand dollars".

MEMORY FUNDAMENTALS

Although the CPU (the computer's brain) can think, it can't remember anything. It can't even remember what problem it was working on!

Besides buying a CPU, you must also buy *memory chips*, which remember what problem the CPU was working on. To find out what the problem was, the CPU looks at the memory chips frequently—about a million times every second!

The part of the main circuitry that contains the memory chips is called the *main memory*.

The typical memory chip comes in a DIP that has 8 pairs of legs (16 pins). In a typical microcomputer, the motherboard contains lots of memory chips.

If you buy extra memory chips (so that your computer can remember extra information), and the extra memory chips don't all fit on the motherboard, you must buy an extra PC card to mount them on; that extra card is called a *memory card*. If the memory card comes in a cute little cartridge that you can pop into and out of the computer easily, it's called a *memory cartridge*.

Warning: if you buy a memory chip or card or cartridge, and want to pop it into the computer, turn off the computer's power first. If you forget, and accidentally leave the power on while you're inserting (or removing) the memory, you might wreck your computer!

You need two kinds of memory chips: *RAM* and *ROM*. The *RAM* chips remember information temporarily; the *ROM* chips remember information permanently. Let's begin by looking at RAM chips.

RAM

If a chip remembers information just temporarily, it's called a *random-access memory chip (RAM chip)*.

When you buy RAM chips, they contain no information yet; you tell the CPU what information to put into them. Later, you can make the CPU erase that information and insert new information instead. The RAM chips hold information just temporarily: when you turn the computer's power off, the RAM chips are automatically erased.

Whenever the CPU tries to solve a problem, the CPU stores the problem in the RAM chips, temporarily. It also stores there all instructions on how to solve the problem; the instructions are called the *program*.

If you buy more RAM chips, the CPU can handle longer problems and programs. If the computer doesn't have enough RAM chips to hold the entire problem or program, you must split the problem or program into several shorter ones instead, and tell the CPU to work on each of the short ones separately.

How RAM is measured. A *character* is any symbol you can type on the keyboard, such as a letter or digit or punctuation mark or blank space. For example, the word HAT consists of 3 characters; the phrase MR. POE consists of 7 characters (M, R, the period, the space, P, O, and E). The phrase LOVE 2 KISS U consists of 13 characters.

Instead of saying "character", hungry programmers say *byte*. So LOVE 2 KISS U consists of 13 bytes. If, in the RAM, you store LOVE 2 KISS YOU, that phrase occupies 13 bytes of the RAM.

RAM chips are manufactured by a process that involves doubling. The most popular unit of RAM is "2 bytes times 2 times 2 times 2 times 2 times 2 times 2 times 2 times 2 times 2", which is 1024 bytes, which is called a *kilobyte*. So the definition of a *kilobyte* is "1024 bytes".

Although a kilobyte is exactly 1024 bytes, the following approximations are useful.

A kilobyte is about a thousand bytes.

A kilobyte is about how many characters you see on the screen of a TV computer.

A kilobyte is about **half** *as many characters as you see on the screen of an 80-column monitor.*

A kilobyte is about a **quarter** *as many characters as you get on a typewritten page (assuming the page is single-spaced with one-inch margins and elite type).*

The abbreviation for *kilobyte* is *K*. For example, if a salesperson says the computer has a "64K RAM", the salesperson means the main circuitry includes enough RAM chips to hold 64 kilobytes of information, which is slightly over 64,000 bytes.

A *megabyte* is 1024 kilobytes. Since a kilobyte is 1024 bytes, a megabyte is "1024 times 1024" bytes, which is 1,048,576 bytes altogether, which is slightly more than a million bytes. It's about how much you can fit in a 250-page book (assuming the book has single-spaced typewritten pages). The abbreviation for *megabyte* is *meg* or *M*.

Rows of RAM chips. In a typical microcomputer, the RAM is a row of eight NMOS chips. Those eight chips, working together, hold 64K. So they hold 64 kilobytes, which is slightly more than 64 thousand bytes (since a kilobyte is slightly more than a thousand bytes).

That row of eight chips is called a *64K chip set*. Each chip in that set is called a "64K chip", but remember that you need a whole *row* of those 64K chips to produce a 64K RAM.

You can buy the entire 64K chip set—the entire row—for a total of just $7, from mail-order electronics supply houses. If you buy the 64K chip set at your local computer store instead, you'll pay about $70.

The most popular style of 64K chip is the *TI 4164*. Although that style was invented by Texas Instruments, other manufacturers have copied it.

If your computer is fancy, it has *two* rows of 64K chips. Since each row is a 64K RAM, the two rows together total 128K.

If your computer is *very* fancy, it has *many* rows of 64K chips. For example, your computer might have four rows of 64K chips. Since each row is a 64K RAM, the four rows together total 256K.

On the other hand, if your computer's extremely old, it contains inferior chips: instead of containing a row of 64K chips, it contains a row of 16K chips. The row of 16K chips produces a 16K RAM (which is inferior to a 64K RAM) and costs just $4 (instead of $7). The most popular style of 16K chip is the *TI 4116*.

If your computer has very little RAM, you can try to enlarge the RAM, by adding extra rows of RAM chips to the motherboard. But if the motherboard's already full, you must buy an extra PC card to put the extra chips on. That extra PC card is called a *RAM memory card*.

The IBM PC is an unusual computer that contains an extra chip in each row, so that each row contains 9 chips instead of 8. The row's ninth chip is called the *parity* chip: it double-checks the work done by the other 8 chips, to make sure they're all working correctly!

Recently, computer engineers have invented 256K RAM chips. A row of eight of those chips gives you 256K of RAM altogether. The most popular style of 256K chip is the *TI 41256*. You can buy a row of 256K chips from electronics supply houses for $24. They're used in Apple's Macintosh computer. But instead of charging you $24 per row, your local Macintosh dealer will charge you *several hundred dollars* instead! That's how Macintosh dealers and Apple Computer Company make their profit!

In a typical *maxi*computer, the RAM contains *many* rows of chips, so that the total RAM contains many megabytes.

Here's how much RAM you typically get altogether:

Price of complete computer	Typical quantity of RAM
$10-$40 (toy micro)	16K (16 kilobytes, 16,384 bytes)
$40-$100 (learning micro)	32K (32 kilobytes, 32,768 bytes)
$100-$400 (home micro)	64K (64 kilobytes, 65,536 bytes)
$400-$1,000 (mid-range micro)	128K (128 kilobytes, 131,072 bytes)
$1,000-$4,000 (business micro)	256K (256 kilobytes, 262,144 bytes)
$4,000-$10,000 (supermicro)	512K (512 kilobytes, 524,288 bytes)
$10,000-$40,000 (mini)	1M (1 megabytes, 1,048,576 bytes)
$40,000-$100,000 (supermini)	2M (2 megabytes, 2,097,152 bytes)
$100,000-$400,000 (modest mainframe)	4M (4 megabytes, 4,194,304 bytes)
$400,000-$1,000,000 (mid mainframe)	8M (8 megabytes, 8,388,608 bytes)
$1,000,000-$4,000,000 (large mainframe)	16M (16 megabytes, 16,777,216 bytes)
$4,000,000-$10,000,000 (supercomputer)	32M (32 megabytes, 33,554,432 bytes)

Although that quantity of RAM is typical, *your* computer might not be "typical". Let's see how much RAM the most popular computers contain.

Commodore's most popular computer is called the *Commodore 64*, because it contains 64K of RAM. Commodore's newer and more expensive computer is called the *Commodore 128*, because it contains 128K of RAM.

The Commodore 64 costs $150. The Commodore 128 sells for about $250, because it includes more RAM and also a better keyboard, a better CPU, a better ROM, and better interfacing.

The Apple 2e and 2c typically include 128K of RAM and sell for about $750. The Apple Macintosh includes 512K of RAM and sells for about $1550; a souped-up version, called the *Macintosh Plus*, includes 1M of RAM and many other improvements and sells for about $2500. Atari's newest computer, the *Atari 520 ST*, includes 512K of RAM but sells for just $750.

The IBM PC comes with anywhere from 64K to 640K; but since most programs for the IBM PC require 256K of RAM, you should buy at least 256K.

NMOS RAM versus CMOS. Most RAM chips are NMOS. The prices I quoted you were for NMOS.

If your computer operates on batteries, it uses CMOS instead, which consumes less electricity than NMOS. Unfortunately, CMOS chips cost more than NMOS. A row of 16K chips costs $4 if they're made of NMOS, but costs $12 if CMOS. A row of 64K chips costs $7 if NMOS, but $32 if CMOS.

Cache. In a maxicomputer or superminicomputer, the RAM is divided into two sections. One section is huge, contains many rows of NMOS chips, and is called the *main RAM*. The other section is tiny, contains just a few bipolar chips, and is called the *cache*.

The cache's bipolar chips work much faster than the main RAM's NMOS chips. For example, in IBM's main maxicomputer (the IBM 3081), the NMOS chips retrieve information in 312 nanoseconds, whereas the bipolar chips take only 26 nanoseconds. (A *nanosecond* is a billionth of a second.)

Unfortunately, the cache's bipolar chips are very expensive and hold just a few K. For example, in the typical IBM 3081 the main RAM holds 32M, whereas the cache holds just 64K.

So the cache is a super-high-speed, super-high-priced memory that's small.

If your computer has a cache, the computer notices which information in the main RAM is being used the most frequently and copies that information into the cache, so the CPU can access that information super-quickly.

Dynamic versus static. A RAM chip is either *dynamic* or *static*. If it's dynamic, it stores data for only 2 milliseconds; after the 2 milliseconds, the electrical charges that represent the data dissipate and become too weak to detect. When you buy a PC board containing dynamic RAM chips, the PC board also includes a *refresh circuit*. The refresh circuit automatically reads the data from the dynamic RAM chips and then rewrites the data onto the chips before 2 milliseconds go by. Every 2 milliseconds, the refresh circuit reads the data from the chips and rewrites the data, so that the data stays refreshed.

If a chip is static instead of dynamic, the electrical charge never dissipates, so you don't need a refresh circuit. (But you must still keep the power turned on.)

In the past, computer designers were afraid that the dynamic RAM's refresh circuit wouldn't work, and used static RAM instead. But today, refresh circuits are reliable, and the most popular kind of RAM is dynamic NMOS. For example, the TI 4116, 4164, and 41256 are all dynamic NMOS.

Dynamic RAM is called *DRAM*. So when an engineer says "give me a DRAM", he doesn't mean whiskey, at least not yet!

Static NMOS is still available. CMOS and bipolar are always static.

If a chip remembers information *permanently*, it's called a *read-only memory chip (ROM chip)*, because you can read the information but can't change it. The ROM chip contains permanent, eternal truths and facts, put there by the manufacturer, and it remembers that information permanently, forever, even if you turn off the power.

Here's the difference between RAM and ROM:

RAM chips remember, temporarily, information supplied by you.
ROM chips remember, permanently, information supplied by the manufacturer.

The typical computer includes many RAM chips (which are arranged in rows), but includes just a *few* ROM chips: just 6 ROM chips, or even fewer.

What kind of information is in ROM? In your computer, one of the ROM chips contain instructions that tell the CPU what to do first when you turn the power on. Those instructions are called the *ROM bootstrap*, because they help the computer system start itself going and "pull itself up by its own bootstrap".

In the typical microcomputer, one of the ROM chips contains instructions that help the CPU transfer information from the keyboard to the screen and printer. Those instructions are called the *ROM operating system* or the *ROM basic input-output system (ROM BIOS)*.

In the typical microcomputer, one of the ROM chips contains information that tells the computer how to make each character on the screen out of dots. That information is called the *character generator*.

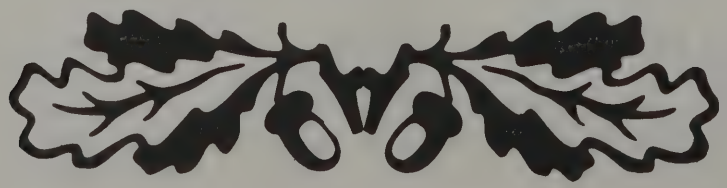
In the typical microcomputer, several ROM chips contain definitions of fundamental English words, which are called *BASIC* words. For example, those ROM chips contain the definitions of BASIC words such as PRINT, NEW, RUN, LIST, GO, TO, END, STOP, INPUT, IF, and THEN. Those definitions in the ROM are called the *ROM BASIC interpreter*.

For example, the typical IBM PC comes with six ROM chips. The first chip contains the ROM bootstrap and ROM BIOS; the second contains the character generator; the remaining four contain the ROM BASIC interpreter. Each of those six chips contains 8K of information, so altogether the IBM PC contains 48K of ROM. (The chip containing the character generator is on a special PC card, called the *video-display card*. The other five ROM chips are on the motherboard.)

The Commodore 64, which is the cheapest popular microcomputer, isn't as fancy as the IBM PC: the Commodore 64 comes with just four ROM chips (instead of six). The first chip contains 8K, for the ROM bootstrap and ROM BIOS; the second contains Commodore's 8K ROM BASIC (which isn't as fancy as IBM's 32K ROM BASIC); the third contains Commodore's 4K character generator (which isn't as fancy as IBM's 8K character generator); and the fourth contains ¼K that tells the computer how to make the screen produce pretty colors. So altogether, the Commodore 64 contains just 20¼K of ROM (whereas the IBM PC contains 48K instead). That's one reason why the Commodore 64 costs less than an IBM PC.

Some microcomputers include extra ROM chips that tell the computer how to handle specific applications, such as word-processing and accounting.

ROM cartridges. If your computer attaches to a TV and is cheap ($50-$500), it lets you insert *ROM cartridges*. A *ROM cartridge* is a cartridge containing a PC card full of ROM chips. Etched into those ROM chips is a program. The typical ROM cartridge contains a program that plays a video game, such as Space Invaders or Pac Man or computer chess. You can also buy ROM cartridges that contain programs for word processing, music, art, or tutoring you. Each ROM cartridge costs about $30.



How ROM chips are made. The information in a ROM chip is said to be *burned into* the chip. To burn in the information, the manufacturer can use two methods.

One method is to burn the information into the ROM chip while the chip's being made. A ROM chip produced by that method is called a *custom ROM chip*.

An alternate method is to make a ROM chip that contains no information but that can be fed information later. Such a ROM chip is called a *programmable ROM chip (PROM)*. To feed it information later, you attach it to a device called a *PROM burner*, which copies information from a RAM to the PROM. Information that's been burned into the PROM can't be erased—unless the PROM is a special kind, called an *erasable PROM (EPROM)*.

To erase the typical EPROM, you shine an intense ultraviolet light at it for 20 minutes. That's called an *ultraviolet-erasable PROM (UV-EPROM)*.

A fancier kind of EPROM can be erased more quickly, by sending it a 25-volt shock for a tenth of a second. That's called an *electrically erasable PROM (EEPROM)* or *electrically alterable PROM (EAPROM)*.

After an EPROM is erased, you can put new information into it.

To design a new computer, the manufacturer begins by using an erasable PROM (EPROM), so that the manufacturer can make changes easily. When the manufacturer decides not to make any more changes, the manufacturer switches to a non-erasable PROM, because it costs less to manufacture. If the computer becomes so popular that over 10,000 copies of the ROM need to be manufactured, the manufacturer switches to a custom ROM, which costs more to design and "tool up for" but costs less to make copies of.

MAIL-ORDER CHIPS

To get chips at discount prices, phone these mail-order suppliers, which are all in California:

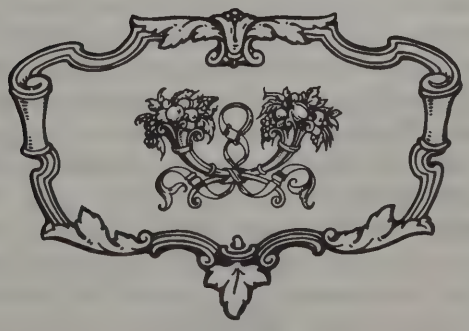
JDR Microdevices, 1224 S. Bascom Ave., San Jose, CA 95128
408-995-5430, out-of-state 800-538-5000

Dokay Computer Products, 2100 De La Cruz Blvd., Santa Clara, CA 95050
408-988-0697, out-of-state 800-538-8800

Jameco Electronics, 1355 Shoreway Rd., Belmont, CA 94002
415-592-8097

Each of them has a $10 minimum. JDR and Dokay charge $2.50 for shipping up to a pound; Jameco charges 5% plus $1.50.

For their latest prices, phone them or read their ads, which appear in *Byte Magazine*. JDR tends to have to lowest prices; Dokay tends to have the highest.



DISKS

APPEARANCE

At the end of the day, when you turn off the computer's power, the computer's RAM forgets everything you told it. Before you turn off the power, tell the computer to examine all the information in the RAM and to copy it all onto a *disk*.

The disk is round. It looks like a phonograph record but has no grooves: it's smooth. It's coated with rust, so that it looks reddish brown. Since the rust is made of iron, which can be magnetized, the disk stores magnetic signals. The magnetic signals are a code that represents your data.

Kinds of disks. Disks come in several sizes: a disk's diameter can be 14 inches, 8 inches, 5¼ inches, or 3½ inches. The most popular size is 5¼ inches, so that the disk looks like a reddish brown 45 RPM record.

Hard versus floppy. You can buy two kinds of disks.

A *hard disk* is sturdy, like a phonograph record.

A *floppy disk* (or *diskette*) is as thin as a sheet of paper. To protect itself from being accidentally being bent and from getting dusty, it comes permanently sealed in a sturdy square *dust jacket*. Never try to remove a floppy disk from its jacket! The jacket has a few holes in it, through which the computer peeks at the disk.

A hard disk holds more bytes of information than a floppy disk, but also costs more to manufacture. A hard disk holds many megabytes and costs several hundred dollars. A typical floppy disk holds about one megabyte—or even less—and costs under $4. Discount dealers sell floppy disks for just 55¢!

Maxicomputers and minicomputers use mainly hard disks. Microcomputers that cost under $2,000 use just floppy disks. Microcomputers that cost between $2,000 and $10,000 use a mixture of floppy disks and hard disks.

Sizes of floppy disks. If the floppy disk's diameter is 8 inches or 5¼ inches, the dust jacket is black. If the floppy disk's diameter is 3½ inches, the dust jacket is traditionally white. (Floppy disks are never 14-inch; that size is only for hard disks.)

Technical note: the jacket of an "8-inch floppy" is exactly 8 inches on each side; the disk's diameter is slightly *less* than 8 inches, so that the disk fits in the jacket. Similarly, the jacket of a "5¼-inch floppy" is exactly 5¼ inches on each side; the disk's diameter is slightly *less* than 5¼ inches.

An 8-inch floppy disk is called a *large floppy*. A 5¼-inch floppy is called a *minifloppy*. A 3½-inch floppy is called a *microfloppy*; it's small enough to fit in the pocket of your shirt.

8-inch floppy disks were invented in the early 1970's by IBM. 5¼-inch floppy disks were invented in the late 1970's by Shugart Associates, which later became part of Xerox. 3½-inch floppy disks were invented in the 1980's by Sony.

8-inch floppy disks have become obsolete, because they're too large to be aesthetically pleasing: the only popular microcomputer that still uses them is the Tandy 6000. Today, 5¼-inch floppy disks are the most popular: they're used by practically every microcomputer company, including IBM, Commodore, Apple, Radio Shack, and the rest of the gang. 3½-inch floppy disks are used in just the newest computers, such as the Apple Macintosh, the Commodore Amiga, the Atari 520 ST, the Tandy 600, the Hewlett-Packard Touchscreen Computer, the Data General 1, and the Apricot; they're also available as an extra-cost option for Apple's older computers (the Apple 2e and Apple 2c).

The typical 3½-inch floppy disk holds 880K (half of that on one side, and the other half on the flip side). An old-fashioned 5¼-inch floppy disk holds just 360K, but the IBM PC AT uses an advanced *high-density* 5¼-inch floppy disk that holds 1200K. The typical 8-inch floppy disk holds about 1200K also.

WHEN TO USE DISKS

After the computer copies the RAM's information onto the disk, you can turn off the computer's power. The information will remain on the disk.

The next time you walk up to the computer (the next day or the next month or even the next century!), you can turn the computer on and tell it to copy information from the disk back to the RAM.

Although the main purpose of disks is to store information when the computer's power is off, disks solve two other problems also.

Problem #1: your computer's RAM isn't large enough to hold all the information you want the computer to handle. Solution: copy the information onto a disk, since the typical disk holds more bytes of information than a typical RAM. If the information's too large to fit on one disk, buy a second disk, or even a third, or more. No matter how large your information is, you can fit it on a collection of disks, and the disks cost hardly anything (as little as 55¢ each).

Problem #2: you want to transfer information from your computer to your friend's computer. Solution: copy the information from your computer's RAM to a disk, and then give the disk to your friend.

FLOPPY DRIVES

To use a floppy disk, you must buy a *floppy-disk drive*, which is a computerized record player.

If the drive is *external*, it's a box sitting near the computer. If the drive is *internal*, it's built into the middle of the computer.

The drive has a slit in its front side. To use the drive, push the disk (including its jacket) into the slit.

When pushing the sheathed treasure into the box's slit, don't shove too hard. Please be gentle!

When you push your disk into the slit, don't push the disk in backwards or upside-down! Here's how to push the disk in correctly. . . .

First, notice that the disk's jacket has a label on it and also has a big oval cutout. Insert the disk so that the oval cutout goes into the drive *before* the label does. If the drive's slit is horizontal, make sure the label is on the *top* side of the jacket; if the drive's slit is vertical, make sure the label is on the *left* side of the jacket.

After putting the disk into the slit, close the latch that covers the slit. The slit and latch act as a *door*; closing the latch is called *closing the door*.

As soon as you close the door, the disk drive automatically positions the disk onto the turntable that's hidden inside the drive. The turntable's called the *spindle*.

The spindle spins the disk quickly. The exact speed depends on what size disk the drive uses. The typical 5¼-inch disk revolves 5 times per second (which makes 300 revolutions per minute, 300 rpm). The 5¼-inch disk in the IBM PC AT revolves slightly faster: 6 times per second (360 rpm). 8-inch disks revolve 6 times per second. 3½-inch disks revolve even faster: between 6½ and 10 times per second.

Like a record player, the disk drive contains an arm with a "needle" on it. The needle is called the *read-write head*, because it can read what's on the disk and can also write new information onto the disk.

To write new information onto the disk, you put your fingers on the computer's keyboard, type a command that tells the computer you want to use the disk, and also type the information you want to transfer to the disk.

To transfer the information to the disk, the computer lowers the read-write head onto the disk. An electrical charge passes through the head. The charge creates an electromagnetic field, which magnetizes the iron on the disk's surface. Each iron particle has its own north and south pole; the patterns formed by the north and south poles are a code that stands for the information you're storing.

Tracks. As the disk spins, the head remains stationary, so that the head draws a circle on the spinning disk's surface. The circle's called a *track*. To draw the circle, the head doesn't use ink; instead, it uses a pattern of magnetic pulses. Since your eye can't see magnetism, your eye can't see the circle; but it's there!

When you start using a disk, the arm puts the head near the disk's outer rim, so that the head's track (circle) is almost as wide as the disk. That track's called *track 0*.

Then the arm lifts the head, moves the head slightly closer to the disk's center, and puts the head back down onto the disk again. The head draws another circular track on the disk, but this new circular track is slightly smaller than the previous one. It's called *track 1*.

Then the head draws track 2, then track 3, then track 4, and so on, until the head gets near the center of the disk, and draws the last circular track (which is smaller than the other tracks).

To organize the information on a track, the computer divides the track into *sectors*. Each "sector" is an arc of the circle.

Single-sided versus double-sided drives. If the disk drive's fancy, it has *two* read-write heads. One head uses the disk's top surface, while the other head uses the disk's bottom, so that the drive can use both sides of the disk simultaneously. That's called a *double-sided disk drive*. The drive puts information onto the disk by first using track 0 of the main side, then track 0 of the flip side, then track 1 of the main side, then track 1 of the flip side, etc.

If the disk drive isn't that fancy, it has just *one* read-write head, which uses just one side of the disk. The flip side of the disk is unused. That kind of drive is called a *single-sided disk drive*. Which side of the disk does the drive use? Though some drives use the side that has the label, other drives (by other manufacturers) use the side *opposite* the label instead.

Capacity. How many kilobytes can you fit on a floppy disk? The answer depends on which computer you have.

The Apple 2e and 2c use 5¼-inch floppy disks in a single-sided drive. Apple's drive puts ¼K per sector, 16 sectors per track, and 35 tracks per disk. So altogether, the disk contains "¼K times 16 times 35", which is 140K.

The Radio Shack Color Computer 2 resembles the Apple 2e, but puts 18 sectors per track instead of 16, so that the disk contains 157½K.

Radio Shack Models 3, 4, and 4P resemble the Radio Shack Color Computer 2, but put 40 tracks per disk instead of 35, so that the disk contains 180K.

The Radio Shack Model 4D resembles the Radio Shack Model 4, but is double-sided instead of single-sided, so that the disk contains 360K.

The IBM PC, PC Junior, Portable PC, and PC XT resemble the Radio Shack Model 4D, but divide each track into 9 fat sectors (½K each) instead of 18 thin ones. The disk still contains 360K.

The Tandy 2000 resembles the IBM PC, but puts 80 tracks per side instead of 40, so that the disk contains 720K.

The IBM PC AT resembles the Tandy 2000, but puts 15 sectors per track instead of 9, so that the disk contains 1200K, which is slightly more than a megabyte (since a megabyte is 1024K).

Other computers are weirder. For example, the Commodore 64 resembles the Apple 2e, but puts an average of about 19½ sectors per track. (The 17 outer tracks each contain 21 sectors; the next 7 tracks each contain 19 sectors; the next 6 tracks each contain 18 sectors; and the inner 5 tracks each contain 17 sectors.) Altogether, the disk contains 683 sectors, which is 170¾K.

The Radio Shack Model 12 and Tandy 6000 use *8-inch* floppy disks. They put ¼K per sector, 32 sectors per track, 77 tracks per side, and 2 sides per disk—except that track 0 of side 1 is deficient: that track uses just 26 sectors, and just an eighth of a K per sector. Altogether, the disk contains 1227¼K.

The Commodore Amiga uses *3½-inch* floppy disks. It puts ½K per sector, 11 sectors per track, 80 tracks per side, and 2 sides per disk, so that the disk contains 880K.

The Apple Macintosh resembles the Commodore Amiga, but puts an average of just 10 sectors per track, and uses just 1 side of the disk, so that the disk contains just 400K. (The 16 outer tracks each contain 12 sectors; the next 16 tracks each contain 11 sectors; the next 16 contain 10; the next 16 contain 9; and the inner 16 contain 8.) A new, improved Macintosh, called the *Macintosh Plus*, uses *both* sides of the disk, so that the disk contains 800K.

Density. Notice that the typical computer uses 5¼-inch floppy disks, and puts between 150K and 200K per side. That kind of computer is said to have a *double-density disk operating system (double-density DOS)*. You get a double-density DOS on the Radio Shack Color Computer 2 (157½K), the Commodore 64 (170¾K), the Radio Shack Models 3 & 4 & 4P (180K), the Radio Shack Model 4D (180K per side), and the IBM PC & PC Junior & Portable PC & PC XT (180K per side). If a DOS puts *less* than 150K per side, the DOS is substandard and is called *single-density*; for example, the DOS on the Apple 2e & 2c is single-density, since it puts only 140K. If a DOS puts *more* than 200K per side, the DOS is super-terrific and is called *high-density*; for example, the DOS on the Tandy 2000 & IBM PC AT is high-density, since it puts 600K per side.

That's how "single-density", "double-density", and "high-density" are defined for 5¼-inch floppy disks. For other size disks, the standards are different.

For *8-inch* floppy disks, "double-density" means "between 500K and 700K per side". For example, the DOS on the Radio Shack Model 12 & Tandy 6000 is double-density, since it puts 613⅝K per side.

The most popular microcomputer for business—the IBM PC—has a double-sided drive, and its DOS is double-density. Its DOS is called *double-sided double-density (DSDD)*.

The Apple 2e & 2c—which are inferior—use a single-sided drive and a DOS that's only single-density. Their DOS is called *single-sided single-density (SSSD)*.

The Commodore 64, Radio Shack Color Computer 2, and Radio Shack Models 3 & 4 & 4P use a DOS that's better than Apple's but worse than IBM's; it's *single-sided double-density (SSDD)*.

BUYING FLOPPY DISKS

When you buy a floppy disk, make sure its size matches the the size of the drive. A 5¼-inch disk will *not* work in an 8-inch drive.

If you buy a blank 5¼-inch floppy disk, you can stick it into any normal 5¼-inch drive, regardless of who manufactured the drive and who manufactured the computer. But after you've put information onto the disk, that information is understandable only to *your* kind of computer. For example, an Apple 2e can*not* understand what an IBM PC writes.

When you go into a computer store to buy a disk that contains software, tell the salesperson which kind of computer you have, so that the salesperson can give you a disk containing information understandable to *your* computer.

Blank 5¼-inch floppy disks are cheap: they cost about $2 each. Although $2 is typical, you can pay anywhere from 55¢ to $6, depending on which brand you buy, which dealer you go to, and which quality level you ask for. If you want to buy an 8-inch or 3½-inch floppy disk instead, you'll pay more, because they're hard to get: only a few companies manufacture them.

Guarantees. The standard 5¼-inch floppy disk is called *double-sided double-density (DSDD)*. It's guaranteed to be free of defects on both sides, and it's guaranteed to be of high enough quality to handle up to 200K per side reliably.

If a 5¼-inch floppy disk is advertised as being "single-sided single-density", the disk comes with less of a guarantee: only one side is guaranteed (the flip side might be defective); only up to 150K per side is guaranteed (squeezing more than 150K onto a side is not guaranteed to be reliable).

If you have an Apple 2e or 2c, whose DOS is just "single-sided single-density", a single-sided single-density disk is good enough for you, since your computer doesn't use the flip side anyway, and since your computer doesn't try to squeeze more than 150K onto a side.

Single-sided single-density disks cost less than double-sided double-density ones, because of the lesser guarantee.

So if you buy an IBM PC and insert a single-sided single-density disk, the disk is not *guaranteed* to work correctly in your computer. But the disk will *probably* work anyway; here's why. . . .

A company that manufactures 5¼-inch floppy disks makes most of them in one big batch. If you insist on double-sided double-density quality, you get a disk that's been thoroughly test and guaranteed, and you pay a high price. If you ask for the lowest quality level (single-sided single-density) instead, you get a disk that's been tested only slightly, and you pay a lower price; that cheap disk is guaranteed to be good enough to hold single-sided single-density information, but the chance is only 95% (not 100%) that it's good enough to hold double-sided double-density or quad-density information: you must test it yourself.

Hobbyists and kids, who have little money, usually ask for single-sided single-density disks, which are cheap, and then test the disks themselves. Business executives, who have lots of money and want insurance against any possibility of erroneous data, usually ask for a guaranteed quality level that matches the computer system: for example, an executive who uses an IBM PC (which is a double-sided double-density system) buys blank disks that are guaranteed to handle double-sided double-density information.

Inner track. The disk's outermost track is the longest; the disk's innermost track is the shortest. The typical DOS squeezes the same number of K onto each track, even though the innermost track is shorter than the others. When the DOS tries to squeeze so many K onto such a short track, errors are likely to occur. And so, in a typical DOS, the innermost track is the one most likely to err.

Commodore's DOS is unusual: it puts fewer K on the inner tracks than on the outer tracks. It therefore makes fewer errors; it works reliably even if you feed it a cheap, single-density disk.

Disks for the IBM PC AT. The IBM PC AT, whose DOS has unusually high density, requires that its disks be manufactured by a different process, in which the disk's coat of magnetic particles is super-thin, super-fine. Standard "double-sided double-density" disks will *not* work on an IBM PC AT; if you try to use such disks, the IBM PC AT will immediately notice your mistake and will print on your screen a message telling you to use a different brand of disks.

Prices. To get the lowest prices on disks, contact *Disk World* and *National Computer Consulting*. For double-sided double-density disks, Disk World charges 69¢ each (if you buy at least 50); National Computer Consulting charges just 55¢ each (if you buy at least 100). Disk World is in Illinois (phone 312-256-7140, out-of-state 800-621-6827); National Computer Consulting is in California (408-496-1022, out-of-state 800-~~425~~ 3111). 245

Although you can buy a blank disk for as little as 55¢, a disk that contains information on it costs much more. The price depends on how valuable the information is. A disk that explains to the computer how to play a game costs about $30. A disk that turns your computer into a personal tutor costs about $40. A disk that explains to the computer how to handle a general business task (such as accounting, filing, or correspondence) usually costs between $100 and $400.

A disk that contains intimate, personal data about your business's customers, suppliers, employees, and methods is worth even more—perhaps *thousands* of dollars! To compute how much it's worth to you, imagine that you've lost it, or that it fell into the wrong hands!

Formatting the disk. After you stick a blank floppy disk into your drive, you must type a command on your computer's keyboard that tells the drive to *format the disk*, which means divide the disk's surface into tracks and sectors.

After the disk has been formatted, you can store whatever information you wish onto the disk. Do *not* tell the drive to format that disk again. If you accidentally make the drive format the same disk again, the drive will create new tracks and sectors on the disk, and erase the old tracks and sectors, and therefore erase all your old data!

Remember . . . If a disk is blank, format it before you use it. If a disk already has information on it, do *not* format it; it's been formatted already.

PROTECTING YOUR DISKS

Most parts of a computer system are sturdy: even if you bang on the keyboard and rap your fist against the screen, you probably won't do any harm. Only one part of a computer system is delicate: that part is the disk. Unfortunately, the magnetic signals on your disk are easy to destroy.

One way to accidentally destroy them is to put your disk near a magnet; so keep your disks away from magnets! For example, keep your disk away from paper clips that have been in a magnetized paper-clip holder. Keep your disk away from speakers (such as the speakers in your stereo, TV, and phone), because all speakers contain magnets. Keep your disk away from electric motors, because motors generate an electromagnetic field. So to be safe, keep your disk at least *six inches* away from paper clips, stereos, TV's, telephones, and motors.

Keep your disk away from heat, because heat destroys the disk's magnetism and "melts" your data. So don't leave your disk in the hot sun; don't leave it on a sunny windowsill; don't leave it in the back of your car on a hot day. If your disk drive or computer feels hot, quickly lower the temperature, by getting an air conditioner or at least a fan.

3½-inch floppy disks come in strong jackets, but 5¼-inch and 8-inch floppy disks come in jackets that are too weak and thin to protect disks from pressure. Don't squeeze your disk. Don't put it under a heavy object, such as a paperweight or a book. If you want to write a note on the disk's jacket, don't use a ball-point pen (which crushes the disk); use a soft felt-tip pen instead.

Keep the disk away from dust. For example, don't smoke cigarettes near the disk, because the smoke becomes dust that lands on the disk and wrecks the data.

Keep the disk dry. If you must transport a disk during a rainstorm, put the disk in a plastic bag. Never drink coffee or soda near the disk: your drink might spill onto the disk.

To handle the disk, touch only the disk's jacket; don't touch the reddish brown disk itself. Holes in the jacket let you see the reddish brown disk inside; don't put your fingers in the holes.

Commodore computers and most computers by Radio Shack (except its IBM PC clones) suffer from this defect: flipping the power switch creates an electrical surge that wrecks the disk. (Apple and IBM computers don't create such a surge.) So on Commodore and most Radio Shack computers, do *not* flip the power switch when the drive contains a disk. Flip the power switch only when the drive's *empty*. To turn on such a computer system, make sure the drive's *empty*, then flip the power switch on; after the power's come on, insert the disk. Before turning such a computer system off, remove the disk from the drive; when the drive's *empty*, turn off the power.

Write-protect notch. When you buy a blank 5¼-inch or 8-inch floppy disk, the disk comes in a square black jacket. Since the jacket's square, it has four sides; but one of the sides has a notch cut into it.

You can cover the notch, by sticking a plastic *tab* over it. The tab has a gummed back, so you can stick it on the disk easily and cover the notch. You get the tab free when you buy the disk.

(For a 3½-inch disk, the notch is different: it's a square hole near the jacket's corner but not on the jacket's edge. To cover it, you use a slider instead of a tab. For Apple Macintosh disks, the slider is red.)

Whenever you ask the computer to change the information on the disk, the drive checks whether you've covered the notch.

For a 5¼-inch disk, the normal situation is for the notch to be uncovered. For a 3½-inch or 8-inch disk, the normal situation is for the notch to be covered.

If the situation's normal, the computer will obey your command: it will change the information on the disk as you wish. But if the situation's *ab*normal (because the notch is covered when it should be uncovered, or is uncovered when it should be covered), the computer will *refuse* to change the disk's information.

Suppose your disk contains valuable information, and you're afraid some idiot will accidentally erase or alter that information. To prevent such an accident, make the situation abnormal (by changing whether the notch is covered), so that the computer will refuse to change the disk's information. It will refuse to erase the disk; it will refuse to add new information to the disk; it will refuse to alter the disk; it will refuse to write onto the disk. The disk is protected from being changed; it's protected from being written on. The disk is *write-protected*.

Since the tab affects whether the disk is write-protected, the tab is called a *write-protect tab*, and the notch is called a *write-protect notch*.

When you buy a disk that already contains information, the disk usually comes write-protected, to protect you from accidentally erasing the information. So if you buy a $5\frac{1}{4}$-inch floppy disk that already contains information, it might come with a write-protect tab already covering the notch, to write-protect the disk.

Instead of creating a notch and then covering it with a tab, some manufacturers save money by getting special disks that have no notch. The computer treats a notchless disk the same way as a disk whose notch is covered.

Backup. Even if you handle your disk very carefully, eventually something will go wrong, and some of the information on your disk will get wrecked accidentally.

To prepare for that inevitable calamity, tell the computer to copy all information from the disk onto a blank disk, so that the blank disk becomes an exact copy of the original. Store the copy far away from the original: store it in another room, or—better yet—another building, or—better yet—another city. If you're working in a country that's having a war, store the copy in another country.

The copy is called a *backup*. Use the backup disk when the original disk gets wrecked.

Making a backup disk is like buying an insurance policy: it protects you against disasters.

Every evening, make backup copies of all your disks—except for disks containing the same information as the day before.

Each week, I get phone calls from distressed business executives whose disks got wrecked and who didn't make backups. All I can offer them is sympathy. Their companies are ruined. Remember: "a backup a day keeps disaster away!"

When you buy a disk that already contains software, try copying the disk before you begin using it. If you're lucky, the computer will make the backup copy without any hassles. If you're unlucky, the software company has put instructions on the disk that make the computer *refuse* to copy the disk, because the company fears that you'll illegally give copies to all your friends for free. A disk that the computer refuses to copy, and which is therefore protected against illegal copying, is called *copy-protected*. A disk that you *can* copy is called *copyable* and *unprotected*.

Short files. The information on the disk is divided into *files*. Each file has its own name and its own purpose. For example, one file might be named JILL, and consist of a memo that you wrote to your friend Jill; another file might be named PAYROLL, and consist of information about your company's payroll. Each file consists of many sectors.

If one of the disk's sectors gets damaged, the computer might get so confused that it handles the entire file incorrectly, and so the entire file becomes unusable.

To minimize such damage, avoid creating large files; create many small files instead. For example, if you're writing a book and want to store the book on your disk, do *not* make the entire book be a single file; instead, split the book into chapters, and make each chapter a separate file. That way, a damaged sector will hurt at most one chapter, and can't hurt the entire book.

HARD DRIVES

Hard disks are better than floppy disks in four ways: hard disks are sturdier, more reliable, hold more information, and work more quickly. Unfortunately, they're also more expensive—and so are their drives.

Hard disks spin very quickly: typically at 60 revolutions per second (3600 rpm).

The most popular sizes for hard disks are 3½-inch, 5¼-inch, 8-inch, and 14-inch. Most microcomputers use 3½-inch or 5¼-inch; most maxicomputers use 14-inch; minicomputers use a mixture of all sizes.

Although the typical floppy-disk drive holds just one disk at a time, the typical hard-disk drive holds a whole stack of disks and handles all of the stack's disks simultaneously, by using many arms and read-write heads.

A hard disk is either *removable* or *non-removable*. If it's *removable*, you can remove it from the disk drive. If it's *non-removable*, you must leave it in the disk drive permanently.

Although floppy disks come in square jackets, removable hard disks come in round *cartridges* or *packs* instead. The typical cartridge or pack contains a whole stack of disks.

How the head works. In a floppy drive, the read-write head (the "needle") touches the spinning floppy disk. But in a *hard* drive, the read-write head does *not* touch the spinning hard disk; instead, it hovers over the disk.

The distance from the read-write head to the hard disk is a tiny fraction of an inch. The distance is small enough so that the read-write head can detect the disk's magnetism and alter it.

Since the head doesn't actually touch the disk, there isn't any friction, and so the head and the disk don't suffer from any wear-and-tear. That's why a hard-disk system lasts longer than a floppy-disk system and is more reliable.

Winchester drives. If your hard drive is ultra-modern, the head acts as a miniature airplane: it *flies* above the disk. It flies at a very low altitude: a tiny fraction of an inch. The only thing keeping the head off the rotating disk is a tiny cushion of air—a breeze caused by the disk's motion.

When you unplug the drive, the disk stops rotating, so the breeze stops, and the head comes to rest on a *landing strip*, which is like a miniature airport.

Such a drive is called a *flying-head drive*. It's also called a *Winchester drive*, because "Winchester" was IBM's secret code-name for that technology when IBM was inventing it.

The head flies at an altitude that's extremely low—about a ten-thousandth of an inch! That's even smaller than the width of a particle of dust or cigarette smoke! So if any dust or smoke lands onto the disk, the head will smash against it, and you'll have a major disaster.

To prevent such a disaster, the entire Winchester drive is sealed air-tight, to prevent any dust or smoke from entering the drive and getting onto the disk. Since the drive is sealed, you can't remove the disks (unless you buy an extremely expensive Winchester drive that has a flexible seal).

Prices. For microcomputers, the most popular hard drive is manufactured by *Seagate*: it uses Winchester technology, contains several non-removable 5¼-inch disks, and holds 20 megabytes altogether. It fits inside the IBM PC, with the help of a PC card containing the disk's controller and manufactured by *Western Digital*. You can buy that 20-megabyte Seagate drive, together with the Western Digital controller card, for $495 altogether, from most mail-order discount dealers. Some dealers even charge as little as $449.

Although most hard drives for microcomputers are 20-megabyte, you can also get other sizes, such as 10-megabyte ($379), 30-megabyte ($695), 40-megabyte, 100-megabyte, and beyond. But the IBM PC is too slow to handle more than 30 megabytes well: to handle so many megabytes, you should get a faster computer, such as an IBM PC AT, and make sure the drive is the high-speed type that the IBM PC AT expects.

For maxicomputers, the typical drive uses Winchester technology, contains a whole stack of non-removable 14-inch disks, holds 800 megabytes altogether, and costs $35,000.

TAPES

TAPES VERSUS DISKS

Like a disk, a magnetic tape consists of magnetized rust. Just as you put a disk into a disk drive, you put a tape into a *tape drive*.

If your computer uses tapes as well as disks, the tapes cost less than the disks, and the tape drive costs less than the disk drive. The main reason people use tapes instead of disks is to save money.

Another advantage of tapes over disks is that since your computer's tapes are smaller and sturdier than the disks, the tapes are easier to stick in an envelope and mail to your friends.

Unfortunately, tape drives are slower than disk drives. To skip from the beginning of a disk to the end of a disk, the disk drive's arm simply hops from the outermost track to the innermost track. But to skip from the beginning of a tape to the end of a tape, you must wait for the tape drive to wind the entire tape.

CASSETTES

The cheapest kind of tape drive is an audio cassette tape recorder—the same kind you use for listening to music, at the beach or in your car. You can attach that kind of tape recorder to the Radio Shack Color Computer 2, which is a cheap TV computer. Wires run from the tape recorder to the computer, and the computer sings a song into the tape recorder; the song is a code that represents the data.

Unfortunately, audio cassette tape recorders aren't very reliable. If you're using a Radio Shack Color Computer 2, you can improve the reliability somewhat by getting Radio Shack's own tape recorder, which is specially designed to work well with computers and automatically controls the tape's volume. But since a tape recorder is so much slower than a disk drive, I recommend that you *not* buy Radio Shack's tape recorder, and instead keep saving your pennies until someday you can afford a disk drive. Once you've experienced the thrilling speed, convenience, and pleasure of a disk drive, you'll never want to use a tape recorder ever again!

Computers by Commodore and Atari do *not* attach to ordinary audio cassette tape recorders; you must buy special cassette tape recorders sold by Commodore and Atari or—better yet—buy a disk drive instead, if you can afford it.

The Coleco Adam computer comes with a built-in cassette tape recorder, at no extra charge. Coleco's tape recorder is high-speed and requires specially lubricated tapes, sold by Coleco. Since it handles just tapes that contain computer information and can*not* play ordinary musical tapes, it's called a *digital cassette tape drive* instead of an audio cassette recorder. But even though Coleco's tape recorder is "high-speed" and handles computer data rather well, it's still not nearly as fast or convenient as a disk drive.

Most people who use 80-column-monitor microcomputers buy two floppy-disk drives and don't bother using tapes at all.

Instead of buying two floppy-disk drives, you might prefer to buy just one floppy drive and then add a Winchester hard drive, which holds more data than a floppy. But how do you make a backup copy of that hard disk, and where do you put the backup? You can't put the backup onto another hard disk, because you bought just one hard drive, which holds just the original non-removable disk. Since you have a floppy drive also, you can put the backup onto floppies; but since each floppy disk holds just a tiny fraction of what the hard disk holds, you'll need a whole box full of floppy disks to backup just one hard disk. As an alternative, some companies sell a special super-fast digital cassette tape drive that holds super-long cassette tapes that can contain backups of the hard disk. But those super-fast tape drives are too expensive and awkward to be popular: use a box of floppy disks instead or buy a second hard drive.

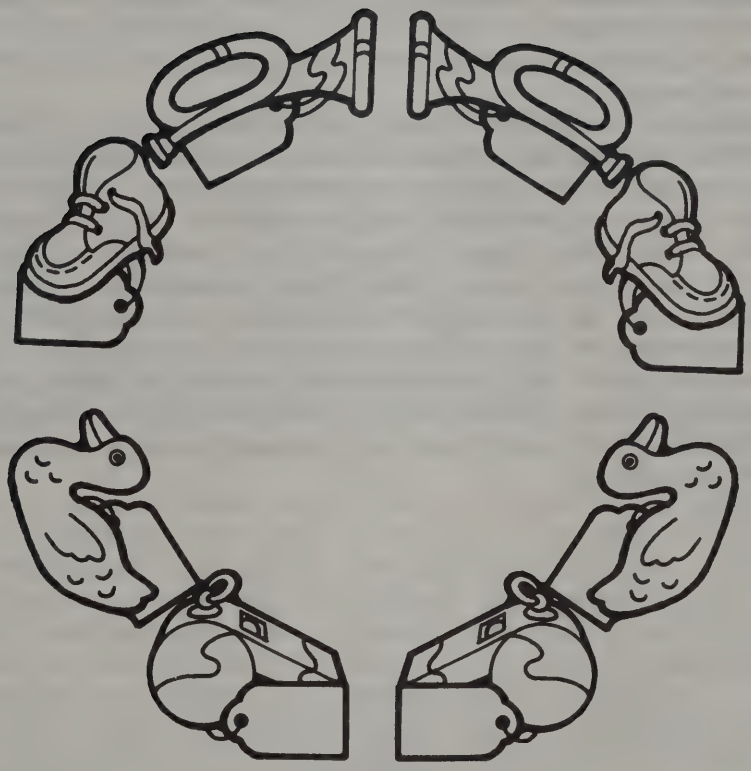
BIG REELS

Maxicomputers and minicomputers use big reels of tape for three purposes: to backup big disks, to send data by mail, and to store the *archives* (old files that are used rarely if ever).

The reel's diameter is $10\frac{1}{2}$ inches. If you unwind the tape, you'll find the tape is half an inch wide and almost half a *mile* long! The exact length is 2400 feet.

To use a reel of tape, you put the reel into a *reel-to-reel tape drive*, which typically costs about $5000 and writes 1600 bytes per inch, so that the entire tape holds 43 megabytes. Super-fancy drives, used only on the largest maxicomputers, squeeze 6250 bytes onto every inch (instead of 1600), so that they squeeze 171 megabytes onto a single reel of tape.

IBM's fanciest drive not only writes 6250 bytes per inch but also does the writing amazingly quickly. It moves the tape at 200 inches per second, so that it transfers about 1.2 megabytes per second.



PRINTERS

FUNDAMENTALS

The typical computer displays its answers on a screen. If you want the computer to copy the answers onto paper, you must attach the computer to a *printer*, which is a device that prints on paper.

The typical printer looks like a typewriter but lacks a keyboard. To feed information to the printer, you type on the *computer's* keyboard.

A cable of wires runs from the back of the computer to the back of the printer. To tell the printer what to print, the computer sends messages through the cable to the printer.

Warning: when you buy a computer, the advertised price usually does *not* include a printer and a cable: they cost extra.

In some ways, printers are worse than screens. Screens are silent; printers are too noisy. Screens work quickly, at electronic speeds; printers are slower. Screens can erase; printers cannot. Screens can show animation; printers cannot. Screens cost less to buy. Screens cost less to operate, since they consume less electricity, no paper, and no ribbons; and they never need repairs.

Although a screen has all those advantages over a printer, you'll want a printer anyway, to put the computer's answers on paper that you can give your computerless friends. Another advantage of paper is that a sheet of paper holds more information than a screen.

Impact printers, like typewriters, print by banging an inked ribbon against the paper. They're very noisy.

Non-impact printers are quieter: they put marks on paper by methods that don't involve banging. The most popular kinds of non-impact printers are *ink-jet printers* (which squirt a jet of ink at the paper), *thermal printers* (whose hot needles scorch the paper), *thermal-transfer printers* (which melt hot colored wax onto the paper), and *laser printers* (which shoot laser beams at a photocopying machine, which then copies the pattern of light onto paper). Each has its own disadvantages: with ink-jet printers, the nozzle that the ink squirts through gets clogged; thermal printers require special "scorchable" paper; thermal-transfer printers require expensive ribbons made of colored wax; and laser printers cost over $2000, which is more than the average microcomputerist can afford. So although non-impact printers are modern, exciting, and quiet, most folks buy impact printers instead, which are just slightly modified typewriters.

TRADITIONAL IMPACT PRINTERS

Impact printers—which bang an inked ribbon against paper—come in several varieties.

The simplest kind to understand is the *daisy-wheel printer*. It contains a *daisy wheel*, which is an artificial daisy flower made of plastic or metal. On each of the daisy's petals is engraved a character: a letter, a digit, or a symbol. For example, one petal has the letter A engraved on it; another petal has B; another petal has C; etc. To print the letter C, the daisy wheel spins, until the C petal is near the inked ribbon; then a hammer bangs the C petal against the ribbon, which in turn hits the paper, so that an inked C appears on the paper.

The typical daisy wheel is plastic rather than metal. Although metal is more durable and provides a sharper image, metal weighs too much and slows down the printer's motion. Another advantage of plastic is that plastic costs less.

You can remove the daisy wheel from the printer, and pop a different daisy wheel into the printer instead. Different daisy wheels contain different *fonts*: for example, one daisy wheel contains italics; a different daisy wheel contains Greek symbols used by scientists.

The printer holds just one daisy wheel at a time. To switch to italics in the middle of your printing, you must stop the printer, switch daisy wheels (an activity which requires your own manual labor and is tedious!), and then press a button that makes the printer resume printing.

The printer can print each character "normally" or "extra-dark". To print a character "extra-dark", the printer prints the character, then moves to the right just 120th of an inch, then prints the same character again, so that the second printing is *almost* in the same place as the original character, and the character looks darkened and slightly fatter. Those darkened, fattened characters are called *boldfaced*.

The most famous manufacturers of daisy-wheel printers are *Diablo* (which is owned by Xerox) and *Qume* (which is pronounced "kyoom"). Both of those companies are American. Recently, Japanese companies have developed daisy-wheel printers that cost less and offer more features. The "best deal" yet comes from Japan: it's the *Panasonic 3151*, which prints beautifully (though slowly), lists for just $659, and is available from discount dealers for just $375.

Another Japanese company, called *Nippon Electric Company (NEC)*, has invented a "wilted" daisy wheel, whose petals are bent. The wilted daisy wheel is called a *thimble*. It's better than a normal daisy wheel: it rotates faster and also produces a sharper image. It's used just in the *Spinwriter* printer and the *Elf* printer, which are both made by NEC. The Elf lists for just $545, and it's available at discount for just $419.

Instead of using a daisy wheel or a thimble, some IBM printers use a plastic *golf ball*, which has characters engraved all over it. IBM calls the golf ball a *Selectric typing element*, and uses it in IBM's *Selectric printers* and Selectric typewriters. Golf balls produce better-looking characters than daisy wheels or thimbles, but rotate more slowly and require more repairs. I used golf balls to print this volume, because I wanted the best-looking characters possible. I used a special souped-up IBM Selectric typewriter attached to a Radio Shack computer.

Gigantic printers—used by maxicomputers and the largest minicomputers—use characters engraved on *bands*, *chains*, *trains*, and *drums*. Those printers can print very quickly, but they cost many thousands of dollars.

IMPACT DOT-MATRIX PRINTERS

Although traditional impact printers use daisy wheels, thimbles, golf balls, bands, chains, trains, and drums, the most popular kind of impact printer uses *pins* instead. When the tip of a pin presses against the ribbon, a dot appears on the paper.

The typical printer holds a column of 9 pins. That column of pins is called the *print head*. If all 9 pins tap the ribbon simultaneously, a column of 9 dots appears on the paper. If only *some* of the pins tap the ribbon, fewer than 9 dots appear on the paper.

Each character is made out of dots. For example, the letter L is made like this:

The left edge of that L contains 7 dots. To produce those 7 dots, 7 of the pins tap the ribbon. Then the column of pins—the print head—moves to the right. To produce the L's bottom, the print head keeps moving to the right as one of the bottom pins repeatedly taps the ribbon.

Although 7 pins are enough to make capital letters (such as A, B, L, and M), 7 pins are *not* enough to make letters that dip below the line (g, j, p, q, and y); for those letters, you need 2 pins "below the line". So altogether, the number of pins you need is 7+2, which is 9. That's why most printers have 9.

Letters that dip below the line are called *descenders*. Before buying a printer, watch it print descenders, to see whether it can print them well. If the printer is crummy, and has only 7 pins instead of 9, it can't print the descenders at all.

Jargon. Each character is made of dots. The pattern of dots that makes up a character is called the *dot matrix*. Printers that make characters out of dots, by tapping 9 pins against a ribbon, are called *9-pin impact dot-matrix printers* or, more briefly, *9-pin impact printers*. Since the pins are also called *wires*, the printers are also called *9-wire impact printers*.

To describe how the pins move, printer manufacturers use military terms. When the pin moves toward the ribbon, the pin is said to be *fired*. The part of the print head that holds the pin, and from which the pin is fired, is called the pin's *gun*. If all 9 pins fire at the ribbon simultaneously, to produce a column of 9 dots, the manufacturer calls it a "9-gun salute".

Epson. The most popular manufacturer of impact dot-matrix printers is *Epson*, which builds about a million printers each month. 40% of the impact matrix printers sold worldwide are built by Epson, and an additional 25% are built by other companies that buy the parts from Epson; so altogether, Epson makes 65% of the impact matrix printer mechanisms worldwide.

Epson became popular because it was the first company to develop a disposable print head—so that when the print head wears out, you can throw it away and pop in a new one yourself, without needing a repairman. Also, Epson was the first company to develop a low-cost impact dot-matrix printer whose dots look "clean and crisp" instead of looking like "fuzzy blobs".

Epson's first 9-pin impact printer was the *MX-80*. Today, Epson's most popular 9-pin impact printers is the *FX-85* ($499 list, $339 at discount). Epson also sells a fancier version, called the *FX-286*, which goes faster and handles wider paper ($749 list, $499 at discount).

Epson is a Japanese company. It's part of a Japanese conglomerate called the *Seiko Group*.

The Seiko Group became famous in 1964: to time the athletes in the 1964 Tokyo Olympics, Seiko invented a quartz clock attached to an electronic printer. Later, the quartz clock was miniaturized and marketed to consumers as the "Seiko watch", which became the best-selling watch in the whole world. The electronic printer, or "E.P.", led to a better printer, called the "son of E.P.", or "EP's son". That's how the Epson division was founded and got its name!

Competitors. Epson's main competitor is *Oki*. Besides building printers, Oki prides itself on having been the first Japanese company to manufacture telephones. Oki's 9-pin impact printer is called the *Okimate*; it comes in several versions.

Besides Epson and Oki, other popular manufacturers of 9-pin impact printers are *Panasonic*, *Citizen*, *Star*, and *C. Itoh*. All those companies are Japanese. I especially recommend the *Panasonic 1091*, because it can print very beautifully (though slowly) and yet costs very little: just $399 list, $222 at discount.

Fonts. The typical impact matrix printer can handle a wide variety of fonts, including italics, boldface, and foreign languages such as Greek and even Japanese! To switch fonts, just tell the computer which font you want, and the computer will make the printer produce it automatically, without requiring you to do any manual labor. It's a lot faster than changing daisy wheels; the printer changes fonts instantaneously!

Beyond 9 pins. Although 9 pins are enough for printing English, 9 pins are *not* enough for printing advanced Japanese, which requires 24 pins instead. Epson, Star, Toshiba, NEC, and several other Japanese manufacturers now sell printers having 24 pins. They print advanced Japanese—and English—beautifully. The cheapest is the *Star SB-10* ($749 list, $479 at discount).

The fastest impact matrix printers use multiple print heads, so that they can print several characters simultaneously.

CHARACTER QUALITY

If your printer uses a daisy wheel, thimble, or golf ball, the characters it produces look about as nice as a typewriter's. A person reading the document might even think it was typed by a human at a typewriter.

But if your printer uses an impact dot matrix instead, each of its characters is made of tiny dots. A person who looks at those dotty characters will realize they were typed by a computer printer, not by a human at a typewriter. A character made of dots looks computerized and impersonal; it's not good enough for a document that tries to look personal.

A printer that prints dots is adequate for sending rough drafts and for bills (though not for personal letters), and so it's called a *draft-quality printer* or a *business-quality printer*. A printer that prints *fully-formed characters* (instead of dots) is called a *letter-quality printer*.

Correspondence quality. If a printer prints dots, but prints the dots so close together that you can hardly tell the dots apart, and so that the characters look *almost* fully formed, and so that the document looks *almost* letter-quality, the printer's called *near-letter-quality (NLQ)* or *correspondence-quality*.

To produce correspondence quality, the typical impact dot-matrix printer begins by printing some dots. Then it jerks the paper up very slightly. The distance it jerks up the paper is so small that it's half as far as the distance between adjoining dots. After jerking the paper, the printer prints more dots, so that the new dots are between the old dots and overlap them. The overlapping dots look very, very similar to a solid line, and so the characters look almost as if they weren't made of dots. The printing, called "correspondence quality", is pretty but slow, since you must wait for the printer to jerk up the paper and put new dots between the old dots.

On the computer's keyboard, you type commands that tell the printer whether you want correspondence quality. If you command the printer to operate in *correspondence mode*, the printer will produce correspondence quality by jerking the paper (slowly); if you command the printer to operate in *draft mode* instead, the printer won't bother to jerk the paper, won't bother to print new dots between old dots, and will produce just draft quality (quickly).

FEEDING THE PAPER

To pull the paper through the printer, the printer can use two methods.

The simplest method is to imitate a typewriter: use a rubber roller that grabs the paper by friction. That method's called *friction feed*. Unfortunately, friction is unreliable: the paper will slip slightly, especially when you get near the bottom of the sheet.

A more reliable method is to use paper that has holes in the margins. The printer has *feeder pins* that fit in the holes. Those pins move vertically and pull the paper up through the printer. The pins move the paper very accurately. That method, which is called *pin feed*, has only one disadvantage: you must buy paper having holes in the margins.

If your printer uses pin feed and is fancy, it has a clamp that helps the pins stay in the holes. The clamp (with its pins) is called a *tractor*. You get a tractor at each margin. A printer that has tractors is said to have *tractor feed*. Usually the tractors are *moveable*, so that you can move the right-hand tractor closer to the left-hand tractor, to handle narrower paper or shorter mailing labels.

A *dual-feed* printer can feed the paper *both* ways—by friction and by pins—because it has a rubber roller and also has sets of pins. The printer has a lever to the left of the roller and pins: if you pull the lever one way, the paper will pass by the roller, for friction feed; if you pull the lever the other way, the paper will pass by the pins, for pin feed.

Paper that has holes in it is called *pin-feed paper* (or *tractor-feed paper*).

When you buy a long tablecloth for your dinner table, the tablecloth comes folded, so you can store it in your closet easily. Likewise, pin-feed paper comes in a long, continuous sheet that's folded. Because it comes folded but can later be unfolded ("fanned out"), it's also called *fanfold paper*. It's perforated, so that you can rip it into individual sheets after the printer finishes printing on it. If the paper's fancy, its margin is perforated too, so that after the printing is done you can rip off the margin, including its ugly holes, and you're left with what looks like ordinary typing paper.

The fanciest perforated paper is called *micro-perf*. Its perforation is so fine that when you rip along the perforation, the edge is almost smooth.

SPEED

The typical advertisement for a printer brags about how fast the printer is, by measuring the speed in *characters per second (cps)* or *lines per minute (lpm)* or *pages per minute (ppm)*. But those measurements are misleading.

The number of "characters per second" depends on which characters are being printed. For example, a daisy wheel will print many "characters per second" if the characters are all the same as each other, so that the daisy never has to rotate. A dot-matrix printer will print many "characters per second" if you're willing to put up with draft quality (instead of near-letter quality, which requires twice as many dots per character). Also, dot-matrix printers are notoriously slow at jerking the paper up to the next line: an "80 character-per-second printer" might print 80 characters within a second, but then take an extra second to jerk the paper up to the next line, so that at the end of two seconds you still see only 80 characters on the paper.

The number of "lines per minute" depends on how many characters are on each line, and depends on which kinds of characters are being printed. Chain, train, and band printers can print more lines per minute if all the letters are capitals; lower-case letters slow them down.

The number of "pages per minute" depends on how big the pages are. Also, you might have to allow for start-up time: a laser printer advertised as printing "8 pages per minute" might print that many only if they're all copies of the same original.

Moral: don't trust any advertisements about speed. The only way to find out the printer's speed is to watch it print many kinds of documents (involving short lines, long lines, short pages, long pages, capitals, lower-case letters, draft quality, and near letter quality), while holding a stopwatch.

WIDTH

A traditional printer makes each character a tenth of an inch wide, so that you get 10 characters per inch. That width, which is the same as an old-fashioned typewriter, is called *pica*.

Some printers make the characters narrower, so that you get 12 characters per inch. That width, which is the same as a modern office typewriter, is called *elite*.

Some printers make the characters extra-narrow, so that you get 16½ or 20 characters per inch. Those widths are called *condensed*.

Some printers make the characters extra-wide, so that you get just 5 characters per inch. That width is called *extended*.

Some printers make each character a different width, so that a "W" is very wide and an "i" is narrow; that's called *proportional spacing*. It look much nicer than the other kinds (pica, elite, compressed, and extended) but is more complicated to handle.

If your printer's fancy, it can handle *all* those widths: it can handle pica, elite, compressed, extended, and proportional.

How wide is the paper? Typically, the printer prints in pica (10 characters per inch), and prints 80 characters per line (to make a total of 8 inches of typing). The paper is slightly wider than 8 inches, to allow for margins. If the printer uses friction feed, the paper is 8½ inches wide (which is the same as ordinary typewriter paper); if the printer uses pin feed, the paper is 9½ inches wide, to let the margins be wide enough to include the holes.

If you tell the printer to print condensed characters (16½ per inch) instead of pica, you get 132 characters per line instead of 80.

For a few hundred dollars extra, you can buy a printer that has an extra-wide carriage, so that it can handle extra-wide paper. Whereas a traditional pin-feed printer uses paper that's 9½ inches wide, a pin-feed printer that has an extra-wide carriage can handle paper that's 14⅞ inches wide instead. By printing condensed characters on that wide paper and by making the margins as narrow as possible, some printers (such as the Epson FX-286) can print 256 characters per line.

INTERFACING

A cable of wires runs from the printer to the computer. The cable costs about $40 and is *not* included in the printer's advertised price: the cable costs extra.

One end of the cable plugs into a socket at the back of the printer. The other end of the cable plugs into a socket at the back of the computer. The socket at the back of the computer is called the computer's *printer port*.

If you're curious about what's inside the computer, and you open up the computer, you'll discover what part of the computer's circuitry the printer port is attached to. In a typical computer, the printer port is attached to the motherboard. But in some computers (such as the Apple 2e and the IBM PC), the printer port is attached to a small PC card instead, called a *printer interface card*; when you buy such a computer, the printer interface card is *not* included in the advertised price, and costs $75 extra.

When the computer wants the printer to print some data, the computer sends the data to the printer port; then the data flows through the cable to the printer.

Serial versus parallel. The cable contains many wires. Some of the cable's wires are never used: they're in the cable just in case a computer expert someday figures out a reason to use them. Some of the wires in the cable transmit information about scheduling: they let the computer and printer argue about when to send the data. If the computer's port is *serial*, just one of the wires transmits the data itself; if the computer's port is *parallel*, eight wires transmit the data simultaneously.

Parallel ports are more popular than serial ports, because parallel ports transmit data faster, are more modern, and are easier to learn how to use. Unfortunately, parallel ports handle only short distances: if the printer is far away from the computer, you must use a serial port instead.

When you buy a printer, make sure the printer matches the computer's port. If your computer's port is parallel, you must buy a parallel printer; if your computer's port is serial, you must buy a serial printer instead.

If your computer has *two* printer ports—one parallel, one serial—you can attach the computer to either type of printer; but I recommend that you choose a printer that's parallel, because parallel printers cost less, and because many word-processing programs require that the printer be parallel.

Standard cables. The typical parallel printer expects you to use a cable containing 36 wires. Of the 36 wires, just 8 transmit the data; the remaining wires can be used for other purposes. That 36-wire scheme is called the *industry-standard Centronics-compatible parallel interface*.

The typical serial printer expects you to use a cable containing just 25 wires. Of the 25 wires, just 1 transmits data from the computer to the printer; the remaining wires can be used for other purposes. That 25-wire scheme is called the *recommended standard 232C serial interface (RS-232C serial interface)*.

Weird cables. If you buy an IBM PC and want to attach it to a parallel printer (which expects 36 wires), you'll get a surprise: the IBM's parallel port contains just 25 wires instead of 36! To attach IBM's 25-wire parallel port to a 36-wire parallel printer, IBM sells a weird cable that has 25 wires on one end and 36 wires on the other.

If you buy an Apple 2c or Apple Macintosh or IBM PC Junior or Radio Shack Color Computer 2 or Commodore 64, and want to attach it to a standard serial printer (which expects 25 wires), you'll get a surprise: the computer's serial port contains fewer than 10 wires! You must buy a weird cable that has fewer than 10 wires on one end and 25 wires on the other.

GRAPHICS

On your screen, you can easily create graphics, such as circles, ovals, graphs, and the Mona Lisa. But how can you transfer those graphics from the screen onto paper?

The easiest way is to photograph the screen with an ordinary camera. For best results, use a monitor instead of a TV, make the room dark (photograph at night with the shades down and without a flash), and use a long exposure time.

An alternative is to use an impact matrix printer, which can make the pictures out of little dots. To make the printer do that easily, you must buy a *graphics screen-dump program*, which copies pictures from the screen to the printer. So if you're buying a printer and want it to do graphics, make sure you also get a graphics screen-dump program, and make sure the program is compatible with *your* computer, *your* printer, and with the other graphics programs that *you're* planning to use.

The Macintosh computer has a built-in graphics screen-dump program, which lets you easily copy graphics from the screen to Apple's *Imagewriter 2* printer, which uses an impact matrix. That nice graphics screen-dump program is one reason why the Macintosh and Imagewriter 2 are popular and usually sold together.

EXOTIC I/O DEVICES

INPUT VERSUS OUTPUT

To use the computer, you feed it a question, and then it feeds you back the answer; or you feed it a problem, and it feeds you back a solution; or you feed it some data, and it feeds you back an analysis; or you feed it a command, and it feeds you back an indication that it obeyed the command.

Everything you feed the computer is called the *input*, because it's what you put in. Everything the computer feeds back to you is called the *output*, because it's what the computer puts out.

The input consists of your questions, problems, data, and commands. The output consists of the computer's answers, solutions, analyses, and accomplishments.

To feed the computer your input, you use an *input device*, such as a keyboard. When the computer wants to feed its output to you, it uses an *output device*, such as a screen or a printer.

Input and output are collectively called *I/O*. So the keyboard, the screen, and the printer are three *I/O devices*. You'll also want three other I/O devices: a *speaker*, a *graphics input device*, and a *modem*. Let's see why. . . .

SPEAKERS

To make the computer produce sounds, you must get a *speaker*, which is a miniature version of the speakers in your stereo system. Since the speaker is cheap, most computer manufacturers include the speaker at no extra charge.

Usually the speaker hides inside the keyboard. Some computers use the speaker that's inside your TV or monitor instead.

By using the speaker, the computer will beep at you whenever you make an error. Fancier computers make the speaker play music. The fanciest computers can speak words, by attaching the speaker to a *speech synthesizer*.

GRAPHICS-INPUT DEVICES

If you feed the computer a picture (such as a photograph, drawing, or diagram), the computer will analyze the picture and even help you improve it.

One way to feed the computer a picture is to attach the computer to a camera. But computerized cameras are expensive, and the computer has difficulty interpreting what the camera sees.

A more practical way to feed the computer a picture is to draw the picture yourself by using a pen attached to a computer, so that the computer notices how you move the pen. The five most popular kinds of computerized pens are *light pens*, *touch screens*, *graphics tablets*, *mice*, and *joysticks*.

A *light pen* is a computerized pen that you point at the screen of your TV or monitor. To draw, you move the pen across the screen. Light pens are cheap: prices begin at $20. But light pens are less reliable, less convenient, and less popular than other graphics-input devices.

A *touch screen* is a special overlay that covers the screen and lets you draw with your finger instead of with a light pen. The only popular computer that uses a touch screen is the *Hewlett-Packard Touchscreen Computer*.

A *graphics tablet* is a computerized board that lays flat on your desk. To draw, you move either a pen or your finger across the board. The cheapest graphics tablet is the *Koala pad*. Koala makes versions of it for each of the popular computers: the Radio Shack version lists for about $60, the Commodore version for $100, the Apple version for $125, and the IBM version for $150.

A *mouse* is a computerized box that's about as big as a pack of cigarettes. To draw, you slide the mouse across your desktop, as if it were a fat pen. The cheapest mouse is made by Radio Shack, which charges just $49.95; other manufacturers charge more. If you buy an Apple Macintosh computer (which is expensive!), you get a mouse at no extra charge.

A *joystick* is a box that has a stick coming out the top of it. To draw, you move the stick in any direction (left, right, forward, back, or diagonally), as if you were the pilot of a small airplane. The most popular joysticks are made by *Atari* and *Kraft*. Atari joysticks work only on Atari and Commodore computers. Kraft joysticks are nicer, work on all the popular computers (IBM, Commodore, Apple, and Radio Shack), and are more expensive. The cheapest joysticks are made by Radio Shack: just $12.50!

MODEMS

To let your computer chat with a computer that's far away, attach both computers to telephone lines, by using a black box that turns computer signals into telephone signals, and turns telephone signals back into computer signals.

Turning a computer signal into a telephone signal is called *modulating the signal*. Turning a telephone signal back into a computer signal is called *demodulating the signal*. Since the black box can modulate and also demodulate signals, the box is called a *modulator/demodulator* (or *modem*, which is pronounced "Moe dem").

Acoustic versus direct-connect. You can buy two kinds of modems. The old-fashioned kind is a black box that has big ears on top, so that it can listen to the telephone. Because of its big ears, it's called a *Mickey Mouse modem* or an *acoustic coupler*. It usually costs $120.

The newer kind of modem plugs directly into the phone system, as if it were an answering machine. It doesn't have any ears: it has telephone wires instead. It's called a *direct-connect modem*. It usually costs just $100, and it's cheaper and more reliable than a Mickey Mouse modem. It's more popular than a Mickey Mouse modem, because it's better than a Mickey Mouse modem in every way, except that you can't attach it to pay phones or to phones in hotel rooms.

Kinds of direct-connect modems. A direct-connect modem can be either *external* or *internal*. If it's *external*, it's a box that sits next to your computer. If it's *internal*, it's a PC card that hides inside your computer. Regardless of whether it's external or internal, a wire runs from it to the phone system.

Internal modems cost less than external ones. The cheapest internal modems cost about $70; the cheapest external modems cost about $100. Internal modems are slightly harder to control, since they lack push-buttons and blinking lights.

The Radio Shack model 100 comes with an internal modem, at no extra charge. An internal modem is a popular option for the Radio Shack Model 4P, the IBM PC, and the Apple 2e. If you have an Apple 2c or an Apple Macintosh, you can't stick PC cards inside your computer, so you can't buy an internal modem: the only modems you can buy are external.

If you pay extra, you can get a direct-connect modem that has super-fancy features, such as *auto-dial* (which means it memorizes the other computer's phone number and dials it whenever you please) and *auto-answer* (which means it automatically answers the phone whenever the other computer calls). A direct-connect modem that has many such fancy features is called *smart*. It's the exact opposite of Mickey Mouse!

Speed. The typical modem transmits 30 characters per second. That speed is called *300 baud*.

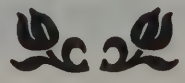
If you pay extra, you can get a faster modem that transmits 120 characters per second. That speed is called *1200 baud*. It's become very popular.

Even faster modems are available (going 2400 baud or even 9600 baud), but they're not cheap enough yet to be popular.

Final choices. Altogether, a smart direct-connect 1200-baud modem costs anywhere from $250 to $600, depending on which brand you buy.

The most popular modems are made by *Hayes*, which charges high prices. Many of Hayes' competitors make cheaper modems that imitate Hayes' and are called *Hayes-compatible*.

You can buy *communications programs*, which help you manage your modem. Most communications programs work only if your modem is by Hayes—or is Hayes-compatible.



SOFTWARE

COMPUTER WARES

A complete computer system consists of three wares: *hardware*, *software*, and *liveware*.

Computer equipment is called *hardware*, because it's built from wires, screws, and other parts you can buy in hardware stores and electronics stores.

The information stored in the computer is called *software*, because you can't feel it: it flows through the computer's circuits in invisible electronic waves. Most software stays in the computer just temporarily, and is erased when you no longer need it. But *some* software stays in the computer's circuits *permanently*: it hides in the ROM and is called *firmware*.

To feed firmware to the computer, stick extra ROM chips into into the main circuity. To feed other kinds of software to the computer, use the keyboard, disk, or tape: type the information on the keyboard, or insert a disk containing the information, or insert a tape.

The person sitting at the computer is called the *operator*, *user*, *liveware*, or *meathead*—because the person's head is made of meat instead of wires. The term *liveware* was invented in 1982 by Gary Trudeau, the creator of Doonesbury cartoons.

Of those three wares—hardware, software, and liveware—software is the most interesting. To make full use of your computer, you must feed it four kinds of software: an *operating system*, a *language*, *applications programs*, and *data*. Let's look at them. . . .

OPERATING SYSTEMS

An *operating system* is a set of instructions that explains to the CPU how to handle the keyboard, the screen, the printer, and the disk drive.

The operating system is divided into two parts. The fundamental part is in the ROM provided by the manufacturer. The advanced part is on a disk and called the *disk operating system* (or *DOS*, which is pronounced "doss"). The disk it comes on is called the *DOS disk*. So to use that advanced part of the operating system, you must put the DOS disk into the disk drive.

Different kinds of computers require different DOS disks. Apple 2e & 2c computers require *Apple DOS* or *Pro DOS*. Radio Shack's Model 4 & 4P & 4D computers require *Tandy's Radio Shack DOS* (which is abbreviated *TRSDOS*, which is pronounced "triss doss"). The IBM PC computer requires either *PC-DOS* or *MS-DOS* (pronounced "em ess doss") or a DOS called *CP/M-86*. Kaypro 2 & 2X & 10 computers require *CP/M-80* instead.

IBM's largest maxicomputers require a DOS called the *Virtual Machine operating system (VM)*.

AT&T's Bell Laboratories invented an operating system called *Unix* (pronounced "you nix") for DEC PDP-11 minicomputers. AT&T has made Unix available for maxicomputers and microcomputers also.

Microsoft has invented a variation of Unix called *Xenix* (pronounced "zee nix"). It runs on the Tandy 6000 computer and the IBM PC.

If a computer imitates a different computer, and the imitation is close enough so that both computers can handle the same DOS, the imitative computer is called a *clone* of the original. For example, many companies sell computers that imitate the IBM PC and can handle the same DOS (PC-DOS or MS-DOS or CP/M-86 DOS); those imitative computers are called *IBM PC clones* or *IBM-compatible microcomputers*.

You can buy IBM PC clones such as the Tandy 1000 & 1200 & 2000 & 3000, the Wang PC, the Texas Instruments Professional, the DEC Rainbow, the computers by Compaq and Leading Edge, and the latest computers by Zenith, Sanyo, Panasonic, AT&T, and ITT. Those clones are all similar enough to the IBM PC so that they can handle the same DOS; but since none of those clones is *exactly* the same as the IBM PC, some of the IBM PC's fanciest software won't run on the clones.

LANGUAGES

Languages that humans normally speak--such as English, Spanish, French, Russian, and Chinese—are called *natural languages*. Unfortunately, natural languages are too complicated for the computer to understand. To communicate with the computer, you must use a *computer language* instead.

The six most popular computer languages are *BASIC*, *LOGO*, *PASCAL*, *C*, *DBASE*, and *COBOL*. Each of those computer languages is a tiny *part* of English—a part that's small enough for the computer to master and totally understand.

To teach the computer one of those computer languages, you feed the computer a ROM or disk containing the definitions of all the words in that language.

In a typical microcomputer, the ROM (provided by the manufacturer) contains the fundamental parts of the operating system and of BASIC. To use the computer fully, you must put into the disk drive a disk that contains the DOS and the advanced parts of BASIC.

Different people prefer to use different languages. Most elementary-school students prefer LOGO; high-school students prefer BASIC; college students prefer PASCAL; advanced programmers using microcomputers and minicomputers prefer C; business executives using microcomputers prefer DBASE; and business executives using maxicomputers prefer COBOL.

Although those six languages are the most popular, many other languages have been invented. Five old languages, which are still used today, are *FORTRAN*, *RPG*, *LISP*, *PL/I*, and *SPSS*. Five new languages, which have their own cult followings, are *FORTH*, *PILOT*, *PROLOG*, *ADA*, and *MODULA*.

The Secret Guide to Computers tutors you in *all* those languages, plus many others, so that you become a virtuoso!

PROGRAMS

The computer will do whatever you wish—if you tell it how. To tell the computer how to do what you wish, you feed it a *program*, which is a list of instructions, written in BASIC or in some other computer language.

To feed the computer a program, type the program on the keyboard, or buy a disk containing the program and put the disk into the disk drive. Before buying a disk that contains a program, make sure the disk will work in *your* computer. For example, if the disk says "for MS-DOS computers", it will work for an IBM PC but not for an Apple.

A person who invents a program is called a *programmer*. Becoming a programmer is easy: you can become a programmer in just a few minutes! Becoming a *good* programmer takes longer.

You can buy two kinds of programs. The most popular kind of program is called an *applications program*: it handles a specific application, such as payroll or psychotherapy or chess. The other kind of program is called a *systems program*: it just helps other programmers write more programs!

An old-fashioned office contains a typewriter, a filing cabinet, and a calculator. A modern office contains a computer instead. To make the computer replace your typewriter, get a *word-processing program*; to replace your filing cabinet, get a *data-management program*; to replace your calculator, get a *spreadsheet program*.

For business-oriented microcomputers (such as the IBM PC), each of those programs comes on a disk. If you're buying a microcomputer to run your business, make sure you also buy those three programs. Tell the salesperson, "I'd like to buy disks containing a word-processing program, a data-management program, and a spreadsheet program."

Since you use only one disk at a time, you need only one disk drive. But the typical business buys a second disk drive also, to more easily store data and to pass data from one program to another.

Why computerize? To save time! By using a word-processing program instead of a typewriter, you can edit mistakes faster. By using a data-management program instead of a filing cabinet, you can locate information without thumbing through file cards. By using a spreadsheet program instead of a calculator, you can rearrange lists and tables of numbers without retyping them.

But I must confess: even in a modern, computerized office, you'll still find a typewriter, a filing cabinet, and a calculator. Those pre-computer relics won't be used much, but they'll still be used *occasionally*, to accomplish tiny tasks for which a computer would be overkill. A typewriter is more practical than a computer, if what you're typing is short (a paragraph or less), or if you're typing answers onto a pre-printed form that was mailed to you. A filing cabinet is more practical than a computer, if you're filing fewer than 100 items, or if you're filing lengthy documents that were mailed to you and that would take too long to retype into the computer. A calculator is more practical than a computer, if you're manipulating fewer than 10 numbers, or if you're writing numbers onto a pre-printed form.

But for *most* tasks, the computer is far superior to pre-computer relics.

Let's take a closer look at what word-processing, data-management, and spreadsheet programs can do for you.

Word processing. A word-processing program helps you write memos, letters, reports, and books. It also helps you edit what you wrote.

As you type on the keyboard, the screen shows what you typed. By pressing buttons, you edit what's on the screen. By pressing another button, the printer prints a copy onto paper. What you typed is also stored on the disk, so that you can make further copies and revisions years later.

The easiest word-processing program is *IBM Writing Assistant*, which runs just on the IBM PC (and clones). It lists for $149.

The fanciest word-processing program is *Word Perfect*, published by *Satellite Software International (SSI)*. It's slightly harder to learn than IBM Writing Assistant; but once you've mastered Word Perfect's intricacies, you'll be thrilled at the powerful ways it helps you edit your writing.

The best version of Word Perfect is called *version 4.1*. It runs just on the IBM PC (and clones). It lists for $495, but most discount mail-order dealers sell it for just $229, and some sell it for as little as $199!

Another version, which isn't quite as fancy, runs on the Apple 2e and 2c. It lists for $179, but discount dealers sell it for just $115.

If you have a Macintosh instead, get *Macwrite* (which is easy) or the Macintosh version of *Microsoft Word* (which is harder but fancier).

Data management. A data-management program helps you manipulate long lists of data, such as names, addresses, phone numbers, birthdays, other information about your friends (and customers and clients and students and teachers and suppliers and employees), past-due bills, and any other data you wish!

As you type the list of data, the computer automatically copies it onto a disk. The computer lets you edit the data and also insert extra data in the middle of the list.

The computer will automatically organize the data and make the printer print the data in any order you wish: in numerical order, or alphabetical order, or chronological order, or ZIP-code order, or however else you please.

The computer will search through all that data and find, in just a few seconds, the data that's unusual. For example, you can tell the computer to find everybody who's birthday is *today*, or to find everybody who has red hair and is under 18, or to find everybody who lives out-of-state and who has owed you more than $30 for over a year.

The easiest data-management program to learn how to use is *PFS File*. The best version (called *version B*) runs on the IBM PC, lists for $140, and is available from discount dealers for just $75. Other versions are available for all the other popular computers, such as the Apple 2e & 2c & Macintosh, the Commodore 64 & 128, and the Radio Shack TRS-80 Models 3 & 4 & 4P & 4D.

After you buy PFS File, you might also want to buy *PFS Report*, which prints tables that summarize the data created by PFS File. The IBM version lists for $125; discount dealers charge $67.

PFS File and PFS Report are both published by *Software Publishing Company*, which plans to publish an even better version of PFS File. The new version will include all the features of the old version plus all the features of PFS Report.

A new company, called *Symantec*, has developed an amazing new program, called *Q&A*, that imitates PFS File, PFS Report, and IBM Writing Assistant. Q&A also performs extra tricks that PFS File, PFS Report, and IBM Writing Assistant can't do. It's the best data-management program available today. Because it does so many extra tricks, it's slightly harder to learn how to use than PFS File. It requires an IBM PC (or clone) with two disk drives and 512K; if you have just one drive or less than 512K or lack an IBM PC, you must use PFS File instead. Q&A lists for $299; discount dealers charge just $219.

Although some other programs perform even fancier tricks than Q&A, those programs are *much* harder to learn how to use; so for most purposes, I recommend Q&A. If you need more tricks than Q&A can provide, your best bet is usually to invent your *own* data-management program, by using *DBASE*, which is a computer language whose only purpose is to let you create your own data-management programs.

The original version of DBASE was called *DBASE 2*; then came an improvement, called *DBASE 3*; then came a further improvement, called *DBASE 3 PLUS*. To run DBASE 3 or DBASE 3 PLUS, you need an IBM PC (or clone). DBASE 2 runs on a wider variety of computers but is inferior.

Spreadsheets. A spreadsheet program handles tables of numbers. The numbers can represent your budget, or your general ledger, or your inventory, or baseball statistics, or students' scores on tests, or whatever else you wish.

As you type the numbers, the computer puts them onto the screen, so that you see neat columns. The computer automatically computes the totals, subtotals, and percentages, and puts them onto the screen also.

Then the computer lets you change the numbers. Whenever you change a number, the computer instantaneously recomputes all the totals, subtotals, and percentages and shows their new values on the screen, faster than your eye can blink!

When the numbers on the screen finally appeal to you (for example, your budget finally balances), you press a button that makes the printer print onto paper the entire table of numbers, including even the totals, subtotals, and percentages. A copy is also stored onto a disk.

The best spreadsheet programs are *Lotus 1-2-3* and *Microsoft Excel*. Lotus 1-2-3 requires an IBM PC, lists for $495, and is available from discount dealers for $283. Microsoft Excel runs on the Macintosh, lists for $395, and is available from discount dealers for $221.

The original version of Lotus 1-2-3 was called *Release 1*. Then came *Release 1A*. Then came *Release 2*.

VP Planner imitates Lotus 1-2-3. But VP Planner lists for just $99, and discount dealers sell it for just $49! Like Lotus-1-2-3, it requires an IBM PC. Although it isn't as good as Lotus 1-2-3 Release 2, it's almost as good as Lotus 1-2-3 Release 1A.

If you have an Apple 2e or 2c, get *Supercalc 3A*, which discount dealers sell for $101. It's almost as good as Lotus 1-2-3 Release 1A.

Integrated programs. Instead of buying a word-processing program and then a data-management program and then a spreadsheet program, you can buy a single "monster" program that does a little bit of everything! Such a program's called an *integrated program*.

The most popular integrated program for the Apple 2e & 2c is *Appleworks*; the most popular for the Apple Macintosh is *Jazz*; the most popular for the Tandy 1000 is *Desk Mate*; the most popular for the IBM PC Junior is *Electric Desk*. For the regular IBM PC, the most popular integrated programs are *Electric Desk*, *Framework*, *Symphony*, *Smart*, *Enable*, and *Ability*.

Unfortunately, each of those integrated programs is a "Jack of all trades, master of none". The typical integrated program does *not* handle word processing as well as Word Perfect, does *not* handle data management as well as Q&A, and does *not* handle spreadsheets as well as Lotus 1-2-3 or Excel. An integrated program is adequate only if you need to do just a *little* bit of everything.

Other applications. Although word-processing programs, data-management programs, and spreadsheet programs are all popular for business, you can also buy programs for art, music, games, education, and many other applications.

DATA DISKS

After you buy a program, your next step is to use it. To do that, put into the disk drive the disk containing the program; that disk is called the *program disk*. After the computer looks at that disk, insert a second disk, called the *data disk*.

If you have two disk drives, put the data disk in the second drive. If you have just one disk drive, put the data disk into that drive, after removing the program disk.

At first, the data disk contains no information; it's blank. But as you put your fingers on the keyboard and start typing the *data* that you want the computer to manipulate, the computer displays that data on the screen and then copies it onto the data disk. So when you've finished using the computer, the data disk contains all the personal data that you fed the computer.

At night, before you go to bed, hide the data disk, so that your enemies can't peek at your personal data.

PAYING FOR SOFTWARE

Altogether, you need four kinds of software: an operating system (a set of instructions that explains to the CPU how to handle the keyboard, the screen, the printer, and the disk drive); a computer language (such as BASIC); applications programs (such as a word-processing program, an electronic spreadsheet, and a data-management system); and data disks (which contains the personal data that you typed yourself).

When you buy a computer, ask the salesperson whether the price includes software. Does the price include the operating system, BASIC, a word-processing program, an electronic spreadsheet, and a data-management system? Usually the advertised price does *not* include all that software; find out how much extra you must pay, to get everything you want.

Before buying the computer system, *try it*, to see whether the hardware and software are easy to use and accomplish your goals pleasantly. Look for quality as well as quantity.

The rest of *The Secret Guide to Computers* explains how to use the computer. It also explains how to judge the quality of hardware and software (so you don't get ripped off), how to invent your *own* software (so that you can make the computer do whatever you please), and how to launch your own computer career.

WOW! YOU FINISHED CHAPTER 2: THE COMPUTER'S PARTS!
TURN THE PAGE FOR CHAPTER 3: BUYER'S GUIDE.

SUBCULTURE

DRUGS

Computers are like drugs: you begin by spending just a little money on them, but then you get so excited by the experience—and so hooked—that you wind up spending more and more money, to feed your habit.

Your first computer experience seems innocent enough: you spend just a little money, for a cute little computer that attaches to your color television. You turn the computer on, and suddenly your television shows dazzling superhuman colors that swirl hypnotically in front of you. You say "Wow, look at all those colors!" and you feel a supernatural high.

But after two months of freaking out with your new computer, the high wears off, and you ask yourself, "What can I buy that's new, exciting, and gives me an even bigger high?" So you buy more stuff to attach to your computer. Now you're in really deep, financially and spiritually. You're hooked. You've become addicted to computers. Each month you return to your favorite computer store, to search for an ever larger high—and spend ever more money.

Look at me, for example: I'm a typical computer junkie. I've already bought 40 computers, and I'm still going. Somebody help me! My computers have already taken over my bedroom, and I've had to create a *second* bedroom, which the computers have already taken over also! Whenever I try to go to sleep, I see those computers staring at me, their lights winking, tempting me to spend a few more hours in naughty fun, even if the sun's already beginning to rise.

Drug addicts easily master computer jargon, because the lingo of today's computerized techno-streets is the same as the lingo of the druggie's hypodermic needle. For example, to buy a computer, you go to a *dealer*; and when you finally start using your computer, you're called a *user*.

As you get in deeper, searching for greater and greater highs, you squander more and more of your money on computer equipment, which is called *hardware*. You stay up late at night (playing computer games, or getting the errors out of your program), so that when you come into work the next morning you're all bleary-eyed. Eventually your boss suspects your computer habit, realizes that you're not giving full attention to your current job, and fires you.

Since you have no job, and your computer bills are getting higher and higher, eventually you run out of money to spend on computers. Yet you still have that urge to spend more. To support your habit, you write programs and try to sell them to your friends. That makes you a "pusher". You turn your friends into addicts too, and you all join the ever-increasing subculture of "computer junkies".

The only difference between computers and drugs is that if you're into drugs people call you a "washout", whereas if you're into computers people say you have a "wonderful career". And they're right!

As a computer "pusher", you can make lots of money. But to be a successful pusher, you must do it in style: instead of calling yourself a "pusher", call yourself a *computer consultant*. Yes, a computer consultant is a person who gives computer advice to other people—and pushes them into buying more computers!

A kind-hearted computer consultant teaches the neighborhood kids about computers, free! But if the consultant seems nice, don't be fooled: the consultant is just a wolf in sheep's clothing. Tom Lehrer wrote a song about it, called "The Old Dope Peddler", which includes these lines:

He gives the kids free samples,
Because he knows full well
That today's young innocent faces
Will be tomorrow's clientele.

That's why Apple is trying to make Congress pass a bill that would let Apple give free samples to every school in America—and get a tax write-off, to boot!

MY MARRIAGE

I'm married to a computer. Marrying a computer is much groovier than marrying a person. For one thing, computers are good at "getting it on": they make you feel all electric and tingly. And computers never argue: they're always ready to "do it" (except when they "have a headache").

I wanted to call this book "The *Sexual* Guide to Computers", and put a picture of me and my wife on the cover; but some parts of the country still prohibit mixed marriages. That cover would be banned in Boston, which (alas!) is where I live. So I had to play it cool, and say "Secret" Guide to Computers. But here's the real secret: this book is about sex.

YOUR MARRIAGE

The computer will fascinate you. It will seduce you into spending more and more time with it. You'll fall in love with it.

You'll even start buying it presents. You'll buy it exotic foods: you'll feed it the most expensive programs to munch on. You'll buy it new clothes: you'll dress it in a pretty little cloth cover, to keep the dust off. You'll adorn it with expensive jewels: a printer and an extra disk drive.

Then the computer will demand you give it more. When you're enjoying an exciting orgy with your computer and think the computer's the best thing that ever happened to you, suddenly the computer will demand that you buy it more memory. It will refuse to let you continue your programming orgy until you agree to its demand. And you'll agree—eagerly!

The computer will be a demanding lover. You'll feel married to it. If you're already married to a human, your human spouse will feel jealous of the computer. Your marriage with your human spouse can deteriorate and end in divorce.

One summer, several women got divorced, because they took my computer course. Their husbands had three complaints:

"You spend most of your time with the computer instead of with me."
"When you do spend time with me, all you want to talk about is the computer!"
"You're not the same woman I married."

To prevent such a marital problem, coax your spouse into playing a video game on the computer. If you're lucky, your spouse will enjoy the game, get hooked on it, and become as addicted to the computer as you. Then your spouse will *enjoy* blabbing about the computer with you and even help you spend money on your habit.

WHY BUY A COMPUTER?

The average American has three goals: to make money, have fun, and "become a better person". Making money is called *business*; having fun is called *pleasure*; and becoming a better person is called *personal development*. The computer will help you meet all three goals: the computer will improve your business, increase your pleasure, and help you to grow into a better person.

The average sucker who buys a computer is a male who comes out at noon. During lunch hour, he walks into a computer store and says he wants to computerize his business. He wants a computer to do his accounting and also handle his mailing list. The computer store's salesperson talks him into also wanting a word-processing program, to help handle business correspondence.

After the sucker visits one or two other computer stores, he buys a computer. Although the computer costs a lot, the salesperson reminds him that Uncle Sam will give a tax break for buying it, since it's a "business expense".

He brings it home, but starts feeling guilty about having spent so much money. How will he convince his wife that the purchase was worthwhile?

Suppose his wife is old-fashioned and spends much of her day cooking. He tries to convince her that the computer will help her cook. "It will help you store your recipes, darling", he coos.

"No thanks", she replies. "When I find a recipe in the newspaper, I don't want to spend 15 minutes typing the entire recipe into the computer. I'd rather just clip the recipe out of the newspaper and—presto! —tape it to a file card. My manual system is faster than a computerized one!"

He tries again. "You could use the computer to store your phone numbers. When you want to look up a phone number, the computer will tell you instantly."

She retorts, "No thanks. To make a phone call, I don't want to have to turn on the computer, find the disk containing the 'telephone' program, put that disk into the disk drive, wait for the computer to ask whose number I'm interested in, sit at the keyboard and type in the jerk's entire name, and then wait for the computer to respond. Instead of doing all that, it's quicker to just open my little black phone book, flip to the page where the number is, and dial my friend. Try again, lover-boy!"

"Well, darling, you could use the computer to remind you of birthdays and appointments."

"You must be crazy! I remember them quite well *without* a computer. I scribble a note on my calendar, which serves fine and costs just $3 instead of $3000. I understand how a disorganized bird-brain male, like *you*, might need a computer to survive; but since we women are better organized, we don't need to rely on mechanical help."

Clearly, the male is making no progress in this conversation, because the computer does *not* fulfill any real need in the home. Nevertheless, he wants to buy it—for a variety of psychological reasons—and searches for an *excuse* to justify the purchase. The computer is a solution looking for a problem.

Of course, women buy computers also. Apple ran a TV ad showing Dick Cavett in a kitchen, as he interviewed a woman who bought a computer. "Oh, you're using it to store your recipes?" he asks.

"No!" she barks. "I'm using it to chart stocks!"

If you buy a computer, the idea of "using the computer to run your business" and "using the computer to store recipes" are just excuses. Here are the *real* reasons why people buy computers. . . .

Teenager: "Computers are a blast: sci-fi come true! Programming computers is the next best thing to becoming an astronaut!"

Parent: "Computers are taking over the world, and my kids will have to learn about them, to survive the coming computer era. The local schools have even made computer courses mandatory. If I buy my kids a home computer, they'll explore it (instead of sex & drugs & rock-and-roll), get curious about how it's programmed, become intellectual programmers, get straight A's in school, grow up to become computer professionals, make lots of money, and share their wealth with me, so I can brag about them to my neighbors."

Grandparent: "I want to be part of the twentieth century. The whole world's becoming computerized, and I don't want my grandchildren to think I'm 'out of it'. I want to share in this new excitement."

Kindergarten kid: "Grandma, please buy me a computer for my birthday! I really want one! Besides, if you don't, the guy on TV says I'll never go to Harvard."

Social climber: "A computer's the only status symbol left for me to buy. Since nobody admires big luxury cars anymore, I'll have to buy a computer instead. And besides, I'm sick of being intimidated by my neighbors, coworkers, and bosses who spout computer jargon. I'm going to learn that mumbo-jumbo myself, so I can get back at those pompous asses and intimidate *them*!"

Worried worker: "My company's starting to use computers. If I don't master those computers, the computers might master *me*, and I'll lose my job! If I learn about computers, I can keep my job and even get a promotion, or—better yet—quit the company and become a rich computer consultant!"

Adventurer: "The computer's a challenge! If I can master the computer, I know I'm not as stupid as people say!"

Middle-aged: "During the past few years, my life's become a bore. I need a new, exciting hobby—like a home computer! Besides being devilish fun, I could use the computer in my business and even turn it into an extra business on the side. And I can keep fiddling with that cute little machine even after my company retires me."

Minister: "Except for the church, the computer's the only thing in our society that has lasting value. Being at peace with computers is like being at peace with God. Besides, with a computer I could double my church membership."

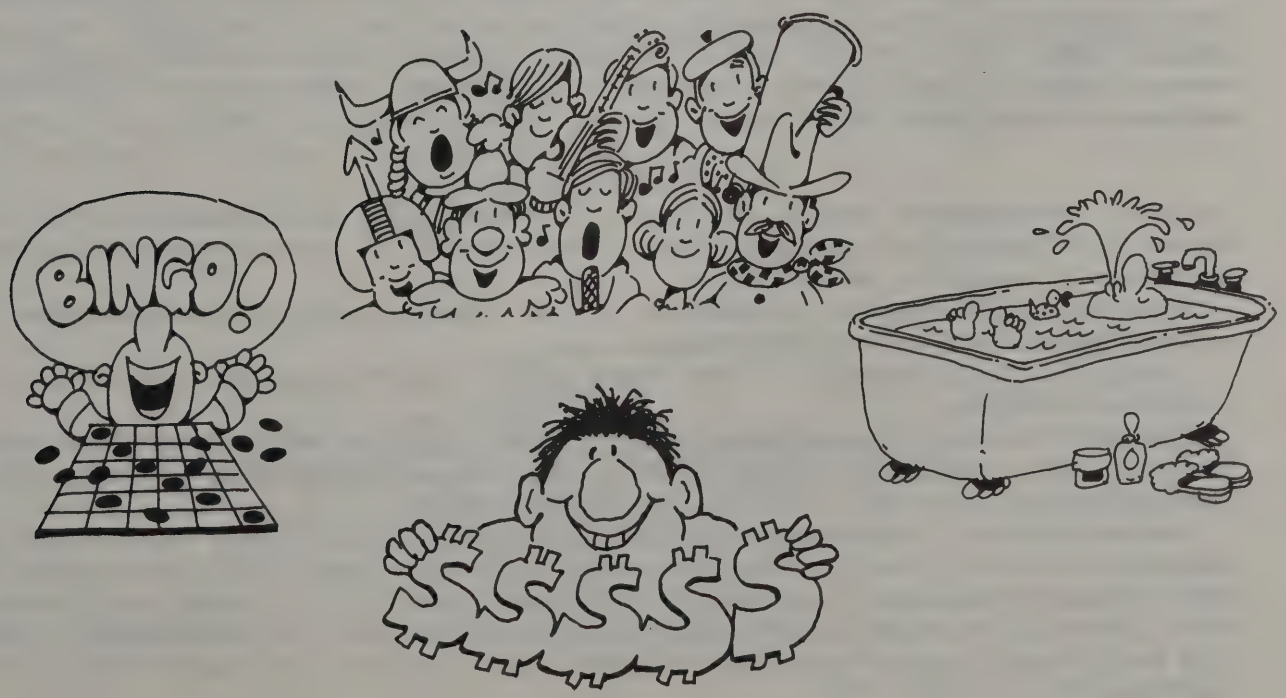
Doctor: "Playing with the computer's anatomy is like playing God. Besides, with a computer I could get my patients to pay their bills!"

Teacher: "I can't believe the way my students pop quarters into those video arcades and get excited about computers too! If only I could get them as excited about poetry! I've got to find out why they love computers, and how to use computers to channel the kids' excitement toward a higher good—such as poetry!"

Proud: "I've worked hard all my life; I *deserve* a computer! I'm gonna get my hands on that mean machine, force it to obey all my commands, and make it my personal slave."

Subversive: "If Big Brother has Big Blue watching *me*, then by gosh I'll turn my computer into Big Mama and scramble their waves!"

Will your computer fulfill all those dreams? Maybe so . . . or maybe not. By the end of this Guide, you'll find out!



PERIODICALS

WHY READ PERIODICALS?

The only way to keep up-to-date about computers is to read newspapers and magazines. They contain the latest computer news, criticize hardware and software, and advise you on what to buy. They also include ads for the newest products and services. I particularly enjoy the ads from discount dealers that reveal their latest low prices.

Some ads and articles use technical computer jargon, which is called *compuspeak*. To understand compuspeak, read The Secret Guide to Computers.

HOW TO BUY PERIODICALS

Visit your local computer stores and bookstores, browse through the computer newspapers and magazines that they sell, and buy one copy of each newspaper and magazine that interests you. (If you live near Boston, visit the kiosks in the middle of Harvard Square. If you live elsewhere, visit your nearest branch of *B. Dalton Bookseller*, which has bookstores in many cities.)

After reading the periodicals you bought, subscribe to the ones you like best. Most periodicals come with a coupon that gives you a "special" discount off the subscription price "for new subscribers, if you hurry". Don't bother hurrying: the same discount is offered to practically everybody every year. And next year, when you renew, you'll be offered the same "special" discount, "for our loyal readers, if you hurry".

Shortly after you pay for a one-year subscription, you'll typically receive a dishonest letter from the publisher warning you that your subscription will "run out soon", and that "if you renew now, you'll get a special discount". Don't believe the letter; "run out soon" usually means "run out eight months from now", and "if you renew now" means "if you renew sometime within the eight months, or even later". Feel free to wait.

HOW TO READ REVIEWS

Many computer newspapers and magazines contain reviews of the newest hardware and software. Read the reviews, but don't take them too seriously: the typical review is written by just one person and reflects just that individual's opinion.

Some reviewers are too easy: they heap praise and say everything is "excellent". Other reviewers are too demanding: they say everything is "terrible". So if one product gets a rave review, and a competing product gets a scathing review, the reason might be the difference between the reviewers rather than the difference between the products.

NEWSPAPERS

The best newspapers about the world of computers are *Computerworld* and *Infoworld*. They're both published once a week by *CW*, a gigantic organization that publishes 57 computer periodicals!

Computerworld covers computers of all sizes. Infoworld concentrates on microcomputers and is more interesting.

Subscribe to Infoworld. If you care about larger computers, subscribe to Computerworld also.

Infoworld costs $1.75 per issue, $31 per year. Computerworld costs $2 per issue, $44 per year officially, $39 per year at discount. To subscribe to both, phone Pennsylvania 215-768-0388, out-of-state 800-544-3712.

Infoworld and Computerworld are both weeklies. For *today's* news about the computer industry, read the business section of your town's daily newspaper, or read national newspapers such as *The Wall Street Journal*, *USA Today*, and *The New York Times*. The computer articles in *The Wall Street Journal* are especially good. The Wall Street Journal costs 50¢ per issue, $107 per year. Its main office is at 22 Cortlandt St. NY, NY 10007.

SPECIFIC MAGAZINES

The best magazine about the IBM PC (and clones) is *PC Magazine*. Besides offering technical details impossible to find elsewhere, it also offers the best comparative reviews of IBM-compatible hardware and software. But since relying on a single magazine is dangerous, you should also get the next-best IBM-oriented magazine, which is *PC World*.

A+ Magazine covers all the computers made by Apple. If you have an Apple Macintosh, you'll also want *Macworld* (the oldest Macintosh magazine) and *Mac User* (a fresh alternative).

Commodore Microcomputers covers all the computers made by Commodore. If you have a Commodore Amiga, you'll also want *Amiga World*.

80 Micro covers all the computers made by Tandy's Radio Shack. If you have a Radio Shack Color Computers, you'll also want *Rainbow*.

If you have a Coleco Adam, sorry, there are no magazines specifically for you! Your best alternative is a newsletter called *Augment*.

Here are the numbers:

PC Magazine, published every two weeks by Ziff-Davis for $2.95. Annual subscription $35 officially, $22 at discount: phone 800-852-5200.

PC World, a monthly by CW for $2.95. Annual subscription $24 officially, $18 at discount: phone Nebraska 402-895-7284, out-of-state 800-628-2900.

A+ Magazine, a monthly by Ziff-Davis for $2.95. Annual subscription $25 officially, $20 at discount: phone Colorado 303-447-9330, out-of-state 800-852-5200.

Macworld, a monthly by CW for $3.95. Annual subscription $30 officially, $24 at discount: phone Nebraska 402-895-7284, out-of-state 800-972-3100.

Mac User, an independent monthly for $3.50. Annual subscription $23 officially, $22 at discount: phone New York 212-302-2626.

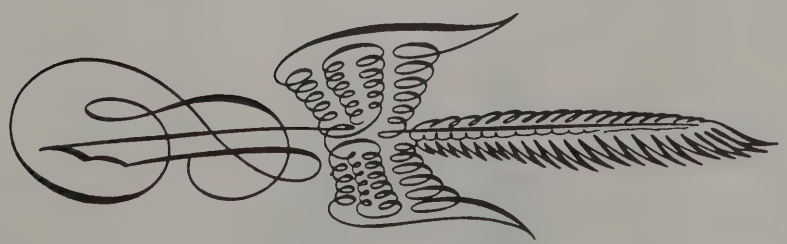
Commodore Microcomputers, published six times a year by Contemporary Marketing for $2.50. Annual subscription $15: phone 800-345-8112; if you're in Pennsylvania, phone 800-662-2444 instead.

Amiga World, published six times a year by CW for $3.95. Annual subscription $20 officially, $15 at discount: phone New Hampshire 603-924-9471, out-of-state 800-258-5473.

Rainbow, a monthly by Falsoft for $3.95. Annual subscription $31: phone Kentucky 502-228-4492.

80 Micro, a monthly by CW for $4. Annual subscription $25: phone New Hampshire 603-924-9471.

Augment, published six times a year by the Adam Users Group for $3. Annual subscription $15: phone New York 516-746-0066.



GENERAL COMPUTER MAGAZINES

The best general computer magazines were *Popular Computing* and *Creative Computing*. But recently, they haven't gotten enough advertisers, and so they've gone out of business. Too bad!

The remaining general computer magazines are *Family Computing* (for beginners using computers under $1000), *Personal Computing* (for beginners using computers over $1000), *Compute* (for experts using computers under $1000), and *Byte* (for experts using computers over $1000). Since The Secret Guide to Computers turns you into an expert, when you finish the Guide you can understand all four magazines.

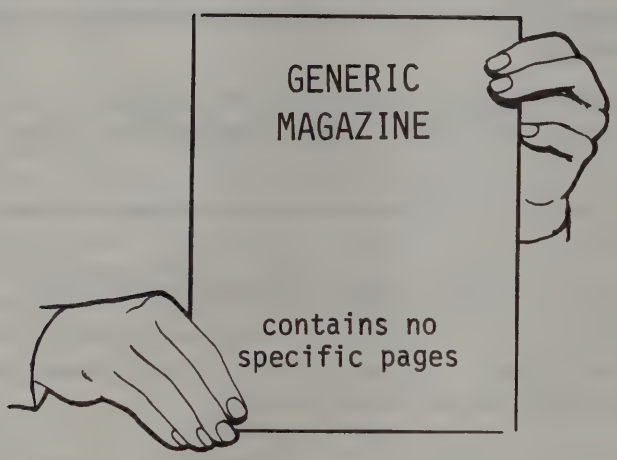
Each of those magazines comes monthly. Here are the numbers:

Family Computing, by Scholastic for $2.50. Subscription $20 officially, $16 at discount: phone 800-525-0643.

Personal Computing, by Hayden for $2.50. Subscription $18 officially, $12 at discount: phone 800-872-3333 extension 41483; if you're in Pennsylvania, phone 800-872-1234 extension 41483 instead.

Compute, by ABC for $2.95. Subscription $24 officially, $18 at discount: phone North Carolina 919-275-9809.

Byte, by McGraw-Hill for $21. Subscription $21: phone New Hampshire 603-924-9281.



DISCOUNT DEALERS

HOW TO BUY AT DISCOUNT

If you browse through the ads in computer magazines and newspapers, you'll see many discount dealers selling all sorts of computer hardware and software, at discounts of 30%, 40%, and even 50%. And if you buy from a dealer who isn't in your state, the dealer won't charge you any sales tax.

Discount dealers change their prices every month. Instead of asking them for catalogs, which might be out of date, peek at their most recent ads, and then double-check those prices by phoning. Usually, prices go down every month, but sometimes they go up instead.

Hardware manufacturers and software publishers improve their products frequently. Before you buy a product from a discount dealer, make sure the dealer has the latest version.

Ask whether the product is in stock, how long the dealer will take to fill your order, and which shipping method the dealer will use. Ask how much the dealer charges for shipping, and whether there's a surcharge for using a credit card.

If the product you ordered turns out to be defective, the dealer or manufacturer will fix it or replace it. But if the product you ordered turns out to be merely "disappointing", and doesn't do as much as you expected or doesn't match the rest of your computer system, tough luck: the typical discount dealer says "all sales are final"; some other discount dealers will let you return the product but will charge you a "restocking fee", which can be up to 25% of the purchase price! So before you buy, ask lots of questions about the product's abilities, so that you're sure it will do what you expect. Tell the dealer what hardware and software you already own, and ask the dealer whether the product is compatible with your system.

RAM cards. A *RAM card* is a PC card that holds RAM chips. If you see an ad for a RAM card, beware: the RAM chips themselves might not be included in the price!

For example, suppose you see an ad for a "384K RAM card". The advertised price might get you a big card that contains no RAM chips on it; the RAM chips might cost extra.

If the card's completely filled with chips, so that you get the full 384K, the card's called *fully populated*. Before you buy a RAM card, ask whether the advertised price is for a card that's fully populated.

Floppy disks. If you see an ad for cheap 5¼-inch floppy disks, make sure the price includes *sleeves* (that you put the disks into when you're not using them) and *labels* (that you stick on the disks and write comments on).

Printers. When you see an ad for a printer, remember that the price usually does *not* include the cable that goes from the printer to your computer. Some printers also need an extra-cost module that customizes the printer to work with *your* computer and attaches the printer to the kind of cable your computer requires. Most salespeople are honest about such matters when you chat with them on the phone, but remember to ask about cables and modules, in case the salespeople forget to tell you.

FAMOUS DISCOUNT DEALERS

Four discount dealers have become famous, because they established new policies. *Conroy-LaPointe* was the first discount dealer to offer a wide variety of hardware and software for all the popular serious computers (the IBM PC, Apple 2e, Apple 2c, and Macintosh). *Micro Connection* was the first discount dealer to advertise friendliness. *Logicsoft* was the first discount dealer to offer overnight express mail shipping at no extra charge and to offer to undercut all competitors by $10. *PC Network* was the first discount dealer to charge customers just "8% over wholesale" and also offer software rentals.

Those four dealers are all successful and reputable—except for PC Network, whose reputation has recently gone downhill (though it might come back up). Lately, dozens of younger dealers have sprung up that offer deals sounding even better, but they're too young to have a track record for reliability yet. The four old famous dealers have remained popular because they've been around long enough so that people trust them. Let's examine them more closely. . . .

Conroy-LaPointe. The "standard" discount dealer is *Conroy-LaPointe*. It sells all the popular hardware and software for the IBM PC, Apple 2e, Apple 2c, and Apple Macintosh. Each Conroy ad gives the list price and discount price for *hundreds* of items. Conroy's discount prices are pleasantly consistent, and set the standard that all the other discount dealers try to compete against.

Look at Conroy's ads. They appear in Infoworld, PC Magazine, A+ Magazine, Macworld, and Byte. Then, if you wish, phone Conroy in Oregon at 503-620-9878, out-of-state 800-547-1289.

After you see Conroy's low price, you must add 3% to it, for shipping.

You can pay by either check or credit card. If you pay by cashier's check or money order, you have no hassle. If you pay by personal check, Conroy waits 20 days, to make sure your check clears. If you pay by credit card, you must pay 3% extra for card-handling (in addition to the 3% extra for shipping).

Charging extra for card-handling is illegal—theoretically. But Conroy skirts the law, by arguing, "Hey, we're *not* charging 3% extra for card-handling! Rather, we give a 3% discount if you *don't* use a card. The prices in our ad assume you're not using a card; so if you're using a card, you must add 3%." Translation: although a card-handling surcharge is illegal, we have a good lawyer, so pay it anyway!

Whenever I want to buy computer hardware and software, I begin by looking at Conroy's ad, which tells the item's list price and standard discount price. Then I check the ads of Conroy's competitors, to find out whether any of them offer prices that are even lower.

In December 1985, Conroy raised its prices slightly, so its prices are not quite as exciting as before. Nevertheless, Conroy is a good place to begin your hunt for the best deal.

Micro Connection. Conroy's largest competitor is *Micro Connection*. It's become even larger than Conroy! Its prices are about the same as Conroy's: sometimes slightly more, sometimes slightly less.

Micro Connection has two divisions.

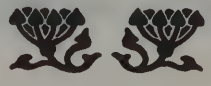
The main division, called *PC Connection*, sells hardware and software for the IBM PC. It advertises in PC Magazine and PC World. Phone New Hampshire 603-446-3383, out-of-state 800-243-8088.

The other division, called *Mac Connection*, sells hardware and software for the Macintosh. It advertises in Macworld and Mac User. Phone New Hampshire 603-446-7711, out-of-state 800-M-A-C-L-I-S-A.

No division's been created yet for the Apple 2e or Apple 2c; so if you have one of those primitive computers, you're out of luck!

Both divisions have excellent reputations for being helpful and offering good service. They even offer technical help at no charge on their toll-free 800 numbers.

Unfortunately, the 800 numbers are usually busy during the day. You stand a better chance if you call in the evening. They're open until 9PM weekdays, 5:30 PM Saturday.



They've been operating out of a barn in the tiny town of Marlow, New Hampshire (whose population is 550, and whose main street consists mainly of a gas station), though they're moving to the inn across the street. They believe in old-fashioned small-town friendliness, and you're welcome to drop in anytime. Adventurers who have trekked to Marlow all report that it's quaint, friendly, and beautiful.

Micro Connection's policies are nicer than Conroy's. Micro Connection waits just one week for personal checks to clear, never charges extra for credit cards, and has low shipping charges: you pay just $2 or 2%.

Logicsoft. Although Conroy's been around for ages and Micro Connection is friendly, I usually wind up buying from *Logicsoft* instead, because Logicsoft has two unusual policies: Logicsoft ships via overnight express mail (Purolator Courier) at no extra charge; and if you find any other ad that offers a lower price than Logicsoft, Logicsoft will undercut that price by $10. Those policies about express mail and undercutting apply only to items that cost over $100.

Logicsoft doesn't charge extra for credit cards.

For shipping, Logicsoft charges 2%. (The headline of Logicsoft's ad says shipping is "free", but the fine print at the bottom of the ad says to add 2% for "handling". I guess in our society, honesty is a lost virtue!)

Although Logicsoft ships via Purolator Courier, Logicsoft sometimes waits a few days before processing the order. (Logicsoft tries to finish processing your order within a day, but doesn't always meet that deadline.) Nevertheless, you'll usually get what you ordered within a week, and often in much *less* than a week. Moreover, Logicsoft has an excellent system for tracking your order's progress.

Phone Logicsoft in New York at 516-249-8440, out-of-state 800-645-3491.

Although Logicsoft undercuts competing prices by $10, the $10-under policy is hard for you to take advantage of, because Logicsoft is smart enough to take its competitors' surcharges into account, and because Logicsoft sometimes requires you to mail a photocopy of the competing ad. But since Logicsoft's advertised prices are already quite low, they'll satisfy you even if you're too lazy to go through the hassle of activating the $10-off policy.

If you're willing to spend half an hour searching through ads in PC Magazine to find the lowest prices, and then activate Logicsoft's $10-off policy, you can save lots of money. For example, look at the ads by the two most aggressive discounters—*Northeastern Software* (Connecticut 203-375-3860, out-of-state 800-382-2242) and *Telemart* (Arizona 602-224-9345, out-of-state 800-426-6659)—and then phone Logicsoft.

Because Logicsoft's become so popular, many other mail-order companies have started to imitate Logicsoft's policies of express mail and price undercutting. For example, Conroy-LaPointe's latest ads brag about shipping via "Federal Express", though the fine print says the shipping is via Federal Express's "standard air" rather than "overnight".

PC Network. The strangest discount dealer is *PC Network*. It sells only to members of its "club", which you can join for $8 per year. Once you're a member, you can buy whatever you wish at "wholesale" prices, plus an 8% service charge, plus shipping. Like Conroy-LaPointe, PC Network supplies a wide variety of hardware and software for the IBM PC, Apple 2e, Apple 2c, and Macintosh.

The term "wholesale" is somewhat misleading, since PC Network's "wholesale" prices are sometimes higher than the regular prices charged by other discount dealers. Nevertheless, even after you add the 8% service charge to the "wholesale" price and the shipping price, PC Network's total price *usually* comes in less than most other discount dealers.

PC Network will also *rent* software to you, for just 20% of the discount purchase price. You can keep the software for a month before returning it. To get full rental privileges, you must pay a membership fee of $45 per year (instead of $8 per year).

Some of PC Network's members, when they rent software, illegally copy it before returning it to the network. Since I dislike illegal activities, I wish PC Network didn't exist: the network is too open to abuse.

But a more serious problem with PC Network is that it treats its own customers dishonestly. Many of the advertised items are not in stock, and PC Network violates the mail-order postal laws, by neglecting to tell customers of the shortages. The Boston Computer Society has received many complaints from consumers who felt ripped-off by PC Network, so I advise you *not* to buy from PC Network until it cleans up its act!

PC Network advertises in PC Magazine, PC World, A+ Magazine, Macworld, Mac User, and Byte. You can reach it in Illinois at 312-280-0002, out-of-state 800-621-S-A-V-E.

NEW YORK JEWS

Two famous discount houses are run by New York Jews. Since they're Jewish and very traditional ("Hassidic"), they're closed on Saturday (the Jewish Sabbath) and Friday afternoon (to prepare for the Sabbath), but they're open on Sunday.

Those discount houses are *Harmony Computers* and *47th Street Computer*.

Harmony Computers. The folks at Harmony first came to my attention several years ago, when I saw them selling Sanyo computers at ridiculously low prices. In those days, Sanyo computers were extremely popular, and I told all my Sanyo-hungry friends to contact Harmony. I never heard any complaints.

Later, when the Atari 800 XL became popular, Harmony again offered the lowest prices. Later, when the Commodore 128 became popular, Harmony again undercut everybody.

Today, Harmony still sells Sanyo, Atari, and Commodore computers at ridiculously low prices. It also sells the Apple 2e, IBM PC, and a wide variety of PC cards, disk drives, monitors, and printers.

Recently, Panasonic printers have become popular, because they've been rated "tops" by Consumer Reports and PC Magazine. Again Harmony has the lowest prices.

Harmony's mission is to find whatever is new and hot, buy huge quantities of it at "distributor" prices, and sell it to consumers for even less than "wholesale". Yes, Harmony charges you even less than your neighborhood dealer's cost!

Harmony operates out of a garage at 2357 Coney Island Ave., Brooklyn, NY 11223. Phone 718-627-1000, out-of-state 800-VIDEO-84 or 800-441-1144. It advertises in Byte, The New York Times, and the Connecticut Computer Society News.

47th Street Computer. This is the computer division of 47th Street Photo, which operates a chain of stores headquartered at 47th Street in Manhattan. The employees are Hassidic Jews who wear traditional black beards, black suits, black coats, and black hats. It sells certain computer hardware and software at ridiculously low prices, because it gets them almost for free, by picking up overstock from other dealers.

47th Street Computer was New York's best-kept secret—New Yorkers would sneak into the store during lunchtime and Sundays and walk out with cheap computers under their arms—until The Wall Street Journal put an article about it on the front page. That made 47th Street Computer famous—and fame breeds higher prices. Today, 47th Street's prices aren't nearly as low as before, but you can still find some bargains. On some overstocked items and other items that were purchased in bulk, 47th Street's prices are unbeatably low; on other items, 47th Street's prices are unremarkable.

I should point out that the New York Department of Consumer Affairs has received more complaints about 47th Street Computer than about any other computer store in New York. But the number of complaints is still within reason, given the many thousands of customers that 47th Street tries to service each week. To avoid feeling "burned", remember that some of the items that 47th Street advertises are not in stock, and that 47th Street's prices change every week, and 47th Street sometimes charges walk-in customers more than the advertised mail-order prices. So before you give 47th Street any money, double-check the price and availability.

RADIO SHACK DISCOUNT DEALERS

Although most Radio Shack stores are owned by the Tandy company, a few Radio Shack stores are independently owned franchises that buy from Tandy at wholesale and charge customers whatever they please. Some of those independently owned franchises offer discounts. To save money, buy from those discount dealers instead of from Tandy-owned stores.

The discount dealer I recommend the most highly is *Computer Plus* (P.O. Box 1094, 480 King St., Littleton, MA 617-486-3193, out-of-state 800-343-8124). It usually offers a 20% discount on hardware, and a 10% discount on software. Its main competitor is *Perry Computers* (137 North Main St., Perry, MI 48872, 517-625-4161, out-of-state 800-248-3823), which offers larger discounts on software. Since Computer Plus and Perry Computers are both authorized dealers, they'll give you an official Radio Shack sales slip and 90-day warranty, entitling you to free repairs at any of Radio Shack's 9000 stores.

DISCOUNTS FROM RETAIL STORES

If you're in a rush for hardware or software and can't wait for mail-order dealers to process your order, look at the ads in the business section of your local newspaper, then visit the local stores and try to talk the salespeople into giving you a discount.

To get a discount, mention the low prices offered by other retail stores within driving distance and the even lower prices offered by mail-order discount dealers and by other retail stores within driving distance, and agree to buy many items at once.

Say that if you don't get a discount, you'll have to shop elsewhere instead. (But say that only if you mean it!)

Most stores allow salespeople to give discounts of up to 10%. But since the discounts are subtracted from the salesperson's commission, you must make a convincing argument!

Stores operated by Radio Shack and IBM do *not* offer such personal discounts, because they consider the discounts unfair to their other customers. If a Radio Shack salesperson is caught in the act of giving a personal discount, the salesperson is fired.

If you live near Boston, join the *Boston Computer Society* (617-367-8080), whose membership card entitles you to a 10% discount at many stores in the Boston area.

If you're planning to buy lots of equipment from Radio Shack, buy a share of Tandy stock, which entitles you to a 10% discount.

IBM, Apple, and Radio Shack often give a 20% discount to schools. The discount's also given to teachers buying computers for use at home, and to students at certain colleges.

But sometimes the discounts are limited to certain schools, certain people, and certain kinds of hardware and software. To find out whether *you* can get any educational discounts, ask your school's administrators. Also ask the IBM, Apple, and Radio Shack stores in your neighborhood.

Radio Shack also offers introductory computer courses to teachers, at no charge!

COMPUTER SHOWS

Another way to get low prices is to attend a computer show. At the typical computer show, many of the booths offer discounts, especially during the last three hours of the last day of the show.

USED COMPUTERS

To pay even less, buy a used computer. The best source of used microcomputers is *The Boston Computer Exchange* (P.O. Box 1177, Boston, MA 02103, 617-542-4414). It gives you, by phone, free information about 1000 used computers you can buy. The exchange has none of the computers in stock; the exchange acts merely as a broker: it passes free information between buyers and sellers of used computers. The exchange collects money only if a seller and buyer agree on a price: in that case, the seller pays the exchange a 10% commission, and the buyer pays the exchange nothing. The exchange gives free advice to sellers about what to charge the buyers, and gives free advice to the buyers about which sellers are offering the lowest prices. The exchange is run by a friendly husband-and-wife team (Alex & Cameron Randall), with the help of many assistants.

What happens if the buyer and seller don't live near each other? Should the seller mail the computer to the buyer, and then hope that the buyer eventually pays for it? Or should the buyer first mail a check to the seller, and then hope that the seller eventually sends the computer?

To solve that problem of "who mails first", the Boston Computer Exchange offers the following deal to both the seller and the buyer, to protect all parties against fraud: first the buyer mails a check to the Boston Computer Exchange; when the exchange receives the check, it tells the seller, who then mails the computer to the buyer; when the buyer receives the computer, the exchange mails a check (minus the 10% fee) to the seller. If the seller neglects to mail the computer, the exchange refunds the check to the buyer. If the seller mails the computer but the buyer dislikes it, the exchange tries to act as a peacemaker, so that both the buyer and the seller wind up happy.

To start your *own* used-computer business, contact the Boston Computer Exchange, which offers franchises all over the world.

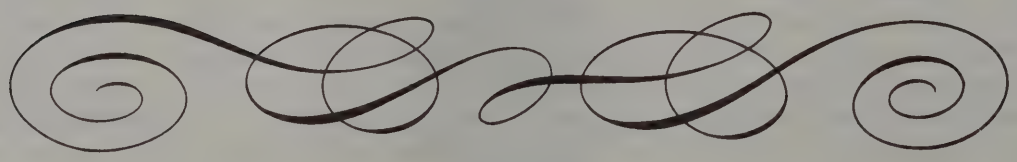
Although the Boston Computer Exchange is the old reliable source of used computers, new competitors have cropped up. For example, contact *Comp-Used*, which resembles the Boston Computer Exchange but charges the seller just 6% (instead of 10%). Comp-Used is at 85 Rivergate Drive, Wilton, CT 06897, 203-762-8677.

REFURBISHED COMPUTERS

If you buy a Commodore computer and you take it in for repair, your local Commodore dealer won't bother to fix it; instead, the dealer will give you a new one. Then the dealer will send your defective computer back to Commodore.

Commodore will repair and refurbish the computer and then sell it to *Comb*, which is a liquidator. Comb resells the refurbished computers to consumers cheaply. For example, Comb has sold refurbished Commodore 64's for $99 (plus $8 shipping), refurbished Commodore disk drives for $149 (plus $8 shipping), and refurbished Commodore printers for $118 (plus $7 shipping).

Comb doesn't have any more Commodore 64's left, but still has some other Commodore equipment in stock. To find out Comb's current prices on refurbished Commodore equipment (and also on Commodore's surplus inventory), contact Comb at 14605 28th Avenue North, Minneapolis, MN 55441, phone 800-328-0609.



COMPANIES TO BE WARY OF

Be very cautious if you try to order goods from PC Network, PC's Limited, or Protecto, even though they all offer low prices that have been praised by magazines. The Boston Computer Society has kept a growing file of complaints about PC Network. Many of the items shipped by PC's Limited are defective, and PC's Limited is negligent in giving refunds or replacements. Protecto sells what its ads say, but its ads are purposely worded to mislead you about what you'll be getting.

Although all of those companies have some happy customers, they also have many customers who are upset. Unless you're willing to gamble, I recommend that you deal with other companies instead.

If you have any particularly happy or sad experiences with any of the mail-order companies I mentioned, or any of their competitors, please tell me; I'm trying to keep track of mail-order reputations. Phone me. If you share your experiences with me, I'll return the favor, by telling you what others have said, and I'll give you an up-to-the-minute analysis of the current mail-order marketplace.

IBM PC

To decide which microcomputer to buy, begin by looking at the IBM PC, because it's the computer that runs the greatest variety of high-quality software. You should seriously consider buying either an IBM PC or a computer that imitates it (an IBM PC clone).

The IBM PC itself is excellent. To invent it, IBM combined all the best ideas that other computer companies had invented previously. IBM did it all legally: IBM found the best hardware and software companies and paid them manufacturing fees and royalties. IBM listened well: IBM put into the IBM PC all the features that business users were begging computer companies to provide, and that could be provided at reasonably low cost.

The only major gripe folks have about the IBM PC is that the SHIFT and ENTER keys are hard to reach. On IBM's new souped-up version (the IBM PC AT), the SHIFT and ENTER keys have been moved slightly closer to the middle of the keyboard.

IBM had originally planned to charge a high price for the IBM PC; but in August 1981, a week before IBM announced the IBM PC to the world, IBM's top management changed its mind, and slashed the price by 25%. So the IBM PC was not only nice but also priced 25% less than the rumor mill had expected. Customers were thrilled and bought IBM PC's quickly.

At first, very few programs were available for it, but IBM turned that liability into a virtue: IBM ran ads telling programmers that since IBM hadn't written enough programs for the PC, programmers could get rich by writing their own. Because of those ads, many programmers bought the PC and wrote thousands of programs for it. All those programs in turn further increased the computer's popularity.

During the past several years, IBM has continually improved the IBM PC's hardware and software and further lowered the price.

How to buy an IBM PC. The IBM PC is a computer that you build yourself: you must buy several separate parts and then put them together. Moreover, each part comes in several different versions, so you can choose the version that best meets your own personal needs.

Software for the IBM PC works best if you buy the following. . . .

Begin by getting what IBM calls a "system unit with 256K and a double-sided drive". It costs $1845. (That's the current list price, but discount dealers charge less, and remember that prices continually drop.) That $1845 gets you the system unit itself (which is a big box), a small keyboard, and a cord that runs from the system unit to the keyboard. Hidden inside the system unit is a motherboard (which contains lots of chips), a double-sided 5¼-inch floppy disk drive, a *disk controller* (which is a PC card that controls the disk drive), and a cable that runs from the disk controller to the disk drive.

On the motherboard, the most exciting chips are the CPU (which is a fancy chip: an Intel 8088), the RAM chips (256K altogether, arranged as four rows of 64K chips), and the ROM chips (40K altogether, arranged as five 8K chips; they contain part of the operating system and part of BASIC).

The motherboard has five slots in it, so that you can insert five PC cards. One of the five slots is already filled: it contains the PC card that's the disk controller. The remaining four slots are empty, so you can buy and insert four more PC cards.

After getting all that for $1845, you'll also want to buy a second double-sided disk drive ($150), a DOS disk (which contains DOS version 2.1 and the most advanced parts of BASIC and costs $65), an RGB color monitor (which IBM sells for $680), a *color card* (a $244 PC card that you attach the color monitor to, and which contains interface chips plus 16K of RAM), and a *printer card* (a $75 PC card that includes a parallel printer port). So altogether, the total list price is $1845+$150+$65+$680+$244+$75, which is $3059. That price includes everything except the printer itself and the printer's cable, which both cost extra.

Although $3059 is the total list price, most dealers offer discounts, so that your total cost is between $2000 and $2500. If even that lower cost is more than you can afford, you can pay less by getting a stripped-down system. For example, you can get a monochrome monitor instead of color, or a TV instead of a monitor, or single-sided drives instead of double-sided, or just one disk drive instead of two, or cassette tapes instead of disks, or get less RAM, or omit the printer port. In fact, if you take *all* those cuts, you can bring the list price down to just $1585: $1265 for a system unit containing just 64K and no drive, plus $244 for a color card, plus $70 for an RF modulator to attach to your home's TV. But if you try any of those cost-cutting measures, some of the software won't work as well.

If a store tries to sell you IBM's monochrome card instead of IBM's color card, beware: not only does the monochrome card lack the ability to handle color, but it also lacks the ability to handle graphics. If you're planning to do any graphics—for art or games or educational software or business charts—you *must* buy the color card (or a clone). On the other hand, if you don't care about graphics, you might prefer the monochrome card, which produces finer looking characters than the color card and gives you a bonus: it lets you underline.

Another way to cut costs is to buy non-IBM parts and insert them into a stripped-down IBM PC. But an even more effective way to pay less is to buy an IBM-compatible microcomputer instead.

MODEL D

Of all the computers that imitate the IBM PC, my favorite is the *Model D*, because it's the cheapest decent computer that closely imitates the IBM PC. My feelings are shared by PC Magazine and Consumer Reports: both of those magazines recommend the Model D more highly than all other clones, because of its high quality and low price.

The Model D was invented in California by Steve Kahng. It's called the "Model D" because it's manufactured by *Daewoo*, which is a huge factory in Korea. The Model D is marketed in the United States by a Massachusetts company, *Leading Edge*, which calls the computer the *Leading Edge Model D*.

The Model D lists for just $1495. That price includes a system unit, a keyboard, an amber TTL monochrome monitor, a DOS disk, and a disk containing a word-processing program.

The system unit includes two double-sided 5¼-inch floppy disk drives, their controller card, and the motherboard. The motherboard includes an 8088 CPU, 256K of RAM, a battery-operated clock (so that the computer doesn't have to ask you the time), 4 slots (one of which holds the disk controller), and 4 ports (a parallel printer port, an RS-232C serial port for attaching a modem, a monochrome port for attaching the monochrome monitor, and a color port for attaching a color monitor).

Although the price includes a monochrome monitor rather than color, you can cheaply add a color monitor, since the motherboard already includes the port for attaching it.

The motherboard's 256K RAM consists of 4 rows of 64K chips. To cheaply increase the RAM to 640K, remove two rows of 64K chips and replace them by 256K chips.

The slots are the same size as the IBM PC's, so that you can insert the same PC cards.

Most of the IBM PC's software will work in the Model D, without any hassles.

The keyboard resembles the one on the IBM PC AT, and so it's better than the one on the plain IBM PC.

Another advantage of the Model D is that its system unit is very compact: it takes up less space on your desk than IBM's, and also weighs less, so that it's easier to carry.

The Model D includes two drives. In the $1495 version, both drives are floppy. In a souped-up version, which costs $1895, one of the drives is floppy but the other drive is hard. The hard drive holds 10 megabytes if your computer was built before March 1986; 20 megabytes afterwards.

Annoyances. Although the Model D costs less than the IBM PC and in many ways is *better* than the IBM PC, it has a few small annoyances. Here they are. . . .

The fan inside the system unit is noisy and sounds like a blast furnace.

To open the system unit (so that you can insert more RAM and PC cards), you need a Phillips screwdriver of exactly the right size.

The RAM chips, which are all on the motherboard, hide underneath the disk drives. If you ever want to change those RAM chips (so that you can expand to 640K), you must first lift the disk drives out of the way.

On the Model D's monochrome monitor, boldface text doesn't look bold enough: it looks too much like plain text.

Although *most* programs for the IBM PC work well on the Model D, a few do not. For example, a data-management program called *Reflex* works on the Model D only if you modify the program slightly; you can get the modification from your Leading Edge dealer or directly from Leading Edge. Other programs that need to be modified for the Model D are *Microsoft Word* (a word-processing program), *Framework* (an integrated package), and *VP Planner* (whose graphics work fine on the Leading Edge's color monitor but not on the monochrome monitor).

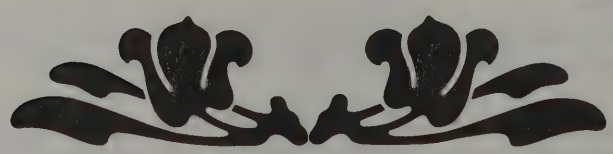
In the IBM PC, most of the BASIC interpreter hides in ROM chips on the motherboard. Those chips are missing from the Model D—and from all other IBM PC clones—because those chips are copyrighted by IBM. For the Model D and other clones, the BASIC interpreter comes on a disk, and is copied from the disk to the RAM.

Where to buy. You can buy the Model D at Computerland (a large chain of stores) and at many small specialty shops. Although Leading Edge illegally threatens to block shipments to any dealer who offers discounts and to any dealer who sells through the mail, some dealers have occasionally offered discounts anyway. In the Boston area, try *Whalley Computer Associates* (617-574-9687 & 617-266-1033), *Bromley Engineering* (617-661-3144), or *The Boston Computer Exchange* (617-542-4414). In Connecticut, try *Electrified* (203-937-0106). In New York City, try *47th Street Computer* (212-608-6934, out-of-state 800-221-7774).

MODEL M

Leading Edge also sells the *Model M*, which is manufactured for Leading Edge by Mitsubishi in Japan. The Model M goes super-fast (50% faster than the IBM PC and the Model D); but because of financial fights between Leading Edge and Mitsubishi, Leading Edge plans to stop selling it.

In December 1985 and January 1986, the Boston Computer Exchange was selling Model M's for just $1000 each. Although that bargain price has ended, you can sometimes get Model M's for under $1200 from the Boston Computer Exchange (at 617-542-4414) and from 47th Street Computer. Although $1000 is an amazingly low price for such a fast computer, remember that the Model M costs more to expand than the Model D, because the Model M's price includes fewer ports and its motherboard holds less RAM.



PC JUNIOR

If you can't afford an IBM PC or Model D or Model M, consider getting IBM's *PC Junior*, which costs less.

The PC Junior comes in two versions. The plain version includes just 64K of RAM and no disk drive and displays just 40 characters per line; the enhanced version includes 128K of RAM and one drive and can display 80 characters per line. The typical system consists of the enhanced version plus an RGB color monitor plus a ROM cartridge containing BASIC plus a disk containing DOS.

IBM has experimented with several prices for the PC Junior, but none of the prices has generated enough profit for IBM. (At high prices, not enough people bought it; at low prices, IBM didn't get enough profit per computer.) So IBM has stopped manufacturing the PC Junior and is offering discounts on the ones that are still in stock.

For the enhanced version plus an RGB color monitor, IBM charges you a total of just $799. Most independent dealers charge even less: just $599!

You'll also need to add $75 for the BASIC cartridge and $65 for the DOS disk, giving a grand total of $599+$75+$65, which is $739.

To pay even less, get a used one from the Boston Computer Exchange.

The Junior costs so little for seven reasons. . . .

1. The price includes just 128K of RAM and one disk drive, whereas most business software requires 256K of RAM and two drives. Although you can add extra RAM and drives to the Junior, the additions are expensive.

2. The Junior includes a serial printer port. Most IBM software assumes you have a parallel printer port instead. Adding a parallel printer port to the Junior costs you $75 extra.

3. The Junior's keyboard contains fewer keys than the PC. The fanciest keys are omitted. Instead of pressing a fancy key, you must press a combination of others keys instead.

4. The Junior is slower than the PC, especially when it tries to write information onto the screen and handle the disk drives.

5. The Junior has fewer slots.

6. The Junior has a smaller power supply (AC/DC transformer). If you try to add several PC cards and cartridges to the PC Junior, you won't have enough electricity, unless you buy an extra power supply.

7. The Junior differs internally from the PC in many little ways, which make the Junior cheaper to manufacture but prevent it from running some of the PC's software and hardware.

Because the Junior costs so little, it's appropriate for young students, for home use, and for very small businesses. But if your needs are somewhat larger, you must buy a senior computer instead.

I especially recommend the Junior for students, because its version of BASIC is great! The Junior's BASIC does everything that the PC can; moreover, in three ways the Junior's BASIC is even *better* than the PC's:

1. The Junior's BASIC can produce multicolored high-resolution graphics.

2. The Junior's BASIC can change the screen's colors instantaneously. For example, you can tell the PC Junior to make everything on the screen that's blue instantaneously change to red.

3. The Junior can produce 3-part musical harmony.

PC XT

Besides the PC and the PC Junior, IBM sells several other microcomputers. The most famous is the *IBM PC Extended (IBM PC XT)*. It resembles the PC, but includes extra slots (for holding extra PC cards) and a larger power supply (to handle those extra cards). It's intended mainly for advanced users who want to add a hard disk.

PC AT

IBM's fanciest microcomputer is the *IBM PC with Advanced Technology (IBM PC AT)*. It's better than the original IBM PC in several ways: its keyboard is laid out better; it has a faster CPU (an 80286 instead of an 8088); it has better floppy disks (high-density instead of just double-density); and it has better hard disks (which hold 30 megabytes and are extra-fast). It can also handle three people simultaneously (on three keyboards and three monitors), though only if you use the Xenix operating system, which is incompatible with most of the popular IBM PC software.

TANDY

Tandy sells four IBM clones: the *Tandy 1000*, the *Tandy 1200*, the *Tandy 2000*, and the *Tandy 3000*. Moreover, each of those clones comes in two versions: the "plain version" (which is cheap) and the *HD version* (which includes a hard disk). Here are the details. . . .

Tandy 1000. The Tandy 1000 is a compromise between the IBM PC and the IBM PC Junior: it's almost as good as an IBM PC, yet costs almost as little as a PC Junior. It's much more compatible with the IBM PC than the Junior is. Critics call it "the computer that the PC Junior should have been".

The Tandy 1000 lists for $999. Unfortunately, that list price does *not* include a monitor ($149.95), does *not* include a second disk drive ($199.95), and does *not* include enough RAM ($239.95 for a card containing 128K of RAM, to add to the 128K of RAM on the motherboard). So altogether, the list price for a complete system is $999 + $149.95 + $199.95 + $239.95, which is $1588.85, which is higher than the list price for a Model D. But since Radio Shack often has a sale where you get an RGB monitor *free*, and since Radio Shack discount dealers offer further discounts of about 20%, you can get a Tandy 1000 for less than a Model D.

The Tandy 1000 imitates the IBM PC more closely than the Junior does, but not as closely as the Model D. The Tandy 1000 differs from the IBM PC in three significant ways. . . .

1. The Tandy 1000's system unit is too short to hold standard IBM PC cards, which are 14 inches long. To fit in a Tandy 1000, a card must be no more than 10 inches long.

2. On the Tandy 1000's keyboard, the keys are arranged in an unusual order. Although Tandy's order is better than IBM's, the difference will confuse you when you're reading manuals about how to use IBM software.

3. The Tandy 1000's video circuitry imitates IBM's color card, but can't imitate IBM's monochrome card. It produces the same ugly characters on a monochrome monitor as it produces on a color monitor.

In spite of those three difficulties, many people buy the Tandy 1000—in fact, it's the best-selling computer that Tandy's ever built! It's more popular than the Model D because you can get it for less and because it's sold and supported by Radio Shack's 9000 stores. Also, its keyboard layout is superior (though different), and its version of BASIC has the same three advantages as the Junior's.

Whereas the Model D's price includes a word-processing program, the price of the Tandy 1000 includes a program called *Desk Mate* instead. Desk Mate consists of six parts: a stripped-down word-processing program, an electronic spreadsheet, a data-management system, a calendar for planning meetings, a communications program, and a program to handle electronic mail. Desk Mate is fine for beginners, though its word-processing program isn't fancy enough for serious business use.

In short, the Tandy 1000 is fine for beginners, though its little incompatibilities annoy advanced users, who often try to buy PC cards intended for full-size system units and buy software intended for IBM keyboards.

Tandy 1200. If you'd rather buy from Tandy than from Leading Edge but can't stand the incompatibilities of the Tandy 1000, consider the *Tandy 1200*. Like Leading Edge's Model D, the Tandy 1200 resembles the IBM PC very closely, without any significant incompatibilities.

The Tandy 1200 lists for $1499, which is about the same price as the Model D. Unfortunately, the Tandy 1200's list price does not include a monitor ($219) or monitor card ($219) or DOS disk ($89.95); so the total price for a Tandy 1200 system is $1499 + $219 + $219 + $89.95, which is $2026.95. On the other hand, you can knock about 20% off that price by contacting Tandy discount dealers, so that the final discount price comes to about the same as the Model D's list price.

Although the Tandy 1200 is fine, the Model D is exceptionally nice, and so I recommend the Model D, unless you live so far from a Leading Edge dealer that you prefer buying from Tandy.

Tandy 2000. The *Tandy 2000* is faster than the IBM PC, though not as fast as the IBM PC AT. It lists for just $1599, plus $199.95 for a monochrome monitor.

Unfortunately, the Tandy 2000's quad-density floppy drives are incompatible with most IBM PC software. Although some IBM PC programs have been converted to the Tandy 2000's drives, and although you can convert some programs yourself, many of the popular IBM PC programs haven't been converted yet and never will.

Tandy hopes that the Tandy 2000's high speed and low price will appeal to artists doing animated graphics, and to draftsmen.

Tandy 3000. If you're thinking of buying an IBM PC AT, you should seriously consider getting the Tandy 3000 instead, which imitates the IBM PC AT very closely but costs much less.

The Tandy 3000 lists for $2599. That price includes 512K RAM, a high-density floppy disk drive, a parallel port, and a serial port. Add $799 for an RGB monitor with card; add $99.95 for a floppy disk containing DOS and Desk Mate; and add $1000 for a 20-megabyte hard disk.

Tandy discount dealers charge even less.

COMPAQ

Although Leading Edge and Tandy sell IBM clones, those companies were not the first. The first IBM clone ever invented was the *Compaq Portable*. It resembled the original IBM PC but was smaller, easier to carry, and included a 9-inch green screen that could display pretty characters (like IBM's monochrome monitor) and yet also display graphics. It became so popular that its manufacturer, Compaq, became the fastest growing computer company ever!

Of all the IBM clones sold today, the Compaq Portable is the safest to buy, because all the major software publishers own Compaq Portables. The publishers all revise their software to make it work on Compaq Portables, before advertising that the software works on "IBM compatibles".

So if you see a program advertised as working on "IBM compatibles", you know it works on Compaq Portables, even though it might not work on the computers by Leading Edge and Tandy.

But because the Compaq Portable is so popular and runs so much software, its manufacturer doesn't need to sell it cheaply, and so the Compaq Portable costs almost the same price as an IBM PC. It's not as aggressively priced as the Model D.

Compaq also manufactures the *Compaq Plus* (which includes a 10-megabyte hard disk), the *Compaq Deskpro* (which has a faster CPU and a larger screen, but isn't portable), the *Compaq Deskpro 286* (which imitates the IBM PC AT), the *Compaq Portable 286* (which imitates the IBM PC AT but weighs less), and the *Compaq Portable 2* (which is a compromise between the Compaq Portable and the Compaq Portable 286).

If you use a Compaq Deskpro computer after dark, you're called a *duskpro*.

When IBM saw that the Compaq Portable was so popular, IBM responded by building the *IBM Portable PC*, which imitates the Compaq Portable, which in turn imitates the IBM PC. You know IBM's in trouble when it stoops so low as to imitate its imitator! Since the IBM Portable PC wasn't quite as good as the Compaq Portable, it didn't become popular. IBM's stopped manufacturing it, but a few are left in stock.

Zenith, which makes televisions, also makes nice clones. Zenith began with the *Z-100*, which was faster than an IBM PC but not compatible enough. Then Zenith developed the the *Z-150*, which imitates the IBM PC closely but isn't as cheap as the Model D. Then Zenith developed the *Z-148*, which is as cheap as the Model D but doesn't contain enough slots for inserting PC cards. Zenith also developed two portables (the *Z-138* and the *Z-160*), which imitate the Compaq Portable. But Zenith's most exciting portable is the *Z-171*: it includes two 5¼-inch disk drives and a screen, yet weighs just 14 pounds altogether and runs on batteries, because it uses CMOS and an LCD screen.

Most of Zenith's computer profits come from contracts with the U.S. government, which buys large quantities of Zenith computers at discount, because Zenith's mastered the art of competitive bidding.

Sanyo made an amazingly cheap IBM clone called the *Sanyo 555-2*. Folks were buying it for just $674 from Harmony Computers (the New York discount dealer). That price included two double-sided disk drives and a collection of software for word processing, spreadsheets, and other tasks. Unfortunately, the price included just 128K of RAM and no monitor. But the computer's main drawback was that it wasn't IBM-compatible enough: half of the popular programs for the IBM PC refused to work on it. Sanyo's stopped making it. You can buy a used one from the Boston Computer Exchange, but make sure you get the 555-2 rather than the 555-1 (whose drives are single-sided) or the 550 (which has just one drive). Sanyo's newer computers—the *675*, *775*, and *885*—are more IBM-compatible but too expensive.

Epson, which makes the world's most popular printers, has developed the *Equity 1*, which imitates the IBM PC very closely. It's priced about the same as the Tandy 1200. Though it costs slightly more than the Model D and includes less circuitry, it's popular because of Epson's excellent reputation as a company.

The *Panasonic Senior Partner* is a portable computer, like the Compaq Portable, but costs less and includes two free extras: a thermal printer and a set of nice business software. Unfortunately, the characters on its screen are somewhat hard to read, and you'll have difficulty inserting PC cards.

AT&T's microcomputer, which is manufactured for AT&T by Olivetti, works fine, but offers no major advantage in cost or features over the IBM PC and IBM PC AT.

The *ITT Xtra* costs even less than the Model D, because ITT is phasing out that model and shifting its efforts to fancier models instead.

The *DEC Rainbow*, the *Texas Instruments Professional*, and the *Wang PC* all offer some improvements over the IBM PC. For example, the DEC Rainbow offers 132 characters per line (instead of 80); the Texas Instruments Professional offers better graphics; and the Wang PC offers a keyboard designed especially for word processing. But those clones are not very IBM compatible: most of the popular programs must be rewritten to work on those computers. Though some popular programs have been rewritten for those computers, the rewrites are sold only by stores that charge the full list price; so buying software for those computers will cost more than buying software for the IBM PC. Wang sells a PC card that makes Wang's computer more IBM-compatible, but that PC card is expensive.

The clones by *Hewlett-Packard* and *Data General* come with floppy drives that are 3½-inch instead of 5¼-inch. To make those clones run IBM software, you must add a 5¼-inch drive and controller, or else wait for software publishers to put their software onto 3½-inch disks.

Inspired by the pretty look of the Hewlett-Packard and Data General clones, IBM invented the *PC Convertible*. It fits in your lap and weighs about 12 pounds. It includes two 3½-inch floppy drives (720K each) and an LCD screen. You can easily detach the LCD screen and switch to a CRT monitor, so that your lap computer converts to a desktop computer; that's why IBM calls it the "Convertible". It runs on batteries, but you can also plug it into the wall or into your car's cigarette lighter.

It costs $1995. That price includes the LCD screen, but not a CRT monitor. It includes 256K of RAM, which you can expand to 512K. The price does *not* include the battery recharger or cigarette-lighter adapter or DOS disk.

Generic clones. Many IBM PC clones sell for under $1000, and some sell for as little as $695! You'll find those low prices advertised in PC Magazine and at computer shows. Those clones aren't manufactured by major companies; instead they're "generic", assembled in American garages from parts manufactured in Taiwan.

Often those low prices do *not* include a monitor, a monitor card, a parallel printer card, a second disk drive, a DOS disk, or a manual. Before you buy a cheap generic clone, ask the dealer what the price includes.

Some of the Taiwanese parts are illegal: those imitations violate the patents and copyrights held by American companies such as IBM, Microsoft (which makes IBM's DOS and BASIC), and Hercules (which makes video cards). So when you see a Taiwanese imitation and ask who manufactured it, the typical dealer refuses to tell you or pretends not to know. (The reputable clones I recommended before—by Daewoo, Mitsubishi, Leading Edge, Tandy, Compaq, Zenith, and the rest of the gang—all differ enough from the IBM PC to avoid lawsuits.)

Many Taiwanese parts are manufactured too hastily and are unreliable. If you buy a Taiwanese clone, you might not find a local dealer willing to repair it.

If you're a novice, you should probably *avoid* those generic clones, because you might get a heart attack from worrying about potential difficulties. But many experts whose nervous systems have become immune to attacks of computer anxiety have no qualms about buying semi-legal clones and taking chances.

If you're thinking of getting a generic clone, first look at the ones by *Kamerman Labs*, *Computer Products Unlimited*, and *PC's Limited*.

For many years, *Kamerman Labs* has been selling disk drives and high-speed tape drives, and has developed a fine reputation. Recently, Kamerman's been selling an IBM PC XT clone for $999. That price includes a floppy drive, a 10-megabyte hard drive, and a monochrome monitor attached to a color card. The price includes only 128K of RAM, but you can increase the RAM to 640K without buying a memory card: just pop extra chips onto the motherboard. The price does *not* include a printer card or a DOS disk or BASIC. Even after you add the cost of the extra RAM chips and the printer card and the DOS disk and BASIC, your total will still be less than if you bought a Leading Edge Model D, and you'll be getting a 10-megabyte hard drive at no extra charge! (On the other hand, the Leading Edge comes with a nicer screen, nicer keyboard, more software, and a larger network of dealers who can provide service and repairs.) Kamerman Labs is in Oregon at 503-626-6877, out-of-state 800-522-2237.

Computer Products Unlimited is another company that offers IBM PC clones at low prices. For example, just $829 gets you the *Computer Products Unlimited System B*, which is a complete computer system, including 640K of RAM, 2 floppy drives, 8 expansion slots, a composite green-screen monochrome monitor attached to a color card, and even a parallel printer port. Just add a DOS disk, a BASIC disk, a printer cable, and a printer, and you're all set! I don't know yet whether the quality and service are good, but if you want to experiment you can phone the company in California at 213-324-3229, out-of-state 800-228-4615.

Many magazines have praised the clones by *PC's Limited*, because those clones have amazingly low prices and yet go even faster than the IBM PC AT. But many consumers *hate* PC's Limited, because much of the computer equipment that the company ships doesn't work: it's *dead on arrival (DOA)*. Moreover, if you try to return the defective computer equipment to PC's Limited for repair or a refund, the company ignores you. Several of my friends who bought from PC's Limited despise the company. Angry complaints about PC's Limited have also appeared in Infoworld (page 16 of the 10/21/85 issue), Computer Update (page 40 of the March/April 1986 issue), and PC World (page 45 of the April 1986 issue).

The president of PC's Limited is a kid named Michael Dell. He started the company when he was 19, in the bedroom of his condo apartment, near the University of Texas in Austin. Now he's 21, and he's taking in millions of dollars each month. He apologizes for the poor service by saying that his company's growing faster than expected, and he can't keep up with the demand for after-sale service. Too bad!

APPLE

APPLE 2E

Apple's first computer was called the *Apple 1*. Then came an improvement, called the *Apple 2*. Then came a further improvement, called the *Apple 2 plus*.

But all those computers are obsolete, because they've been replaced by a further improvement, called the *Apple 2 enhanced (Apple 2e)*, which most stores sell for just $749.

Many programs were written for the old Apples: the Apple 1, the Apple 2, and the Apple 2 plus. Most of those programs also run on the Apple 2e.

Slots. In each old Apple, the motherboard contained eight slots, numbered from 0 to 7. Each slot could hold a PC card.

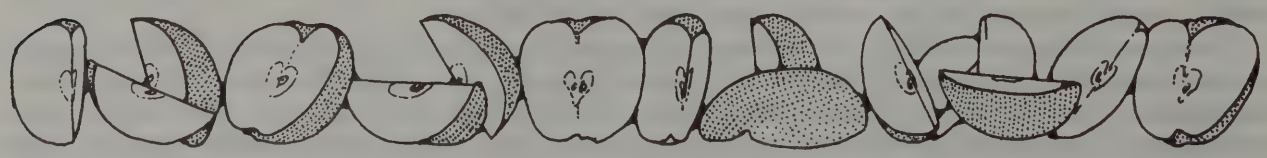
Slot 0 was for a PC card containing extra RAM. Slot 1 was for a PC card containing a parallel printer port. Slot 2 was for a PC card containing a serial printer port or a modem. Slot 3 was for a PC card containing extra video circuitry that made the Apple display 80 characters per line instead of 40. Slot 6 was for a PC card containing a disk controller. Slots 4, 5, and 7 were for PC cards that were more exotic.

The Apple 2e's slots are similar—with two exceptions. The Apple 2e omits slot 0, because the Apple 2e doesn't need a RAM card: the Apple 2e's motherboard already contains lots of RAM (64K). The Apple 2e contains an extra slot, called 3A, which resembles slot 3 but holds a more modern kind of video card that comes in two versions: the plain version lets your Apple display 80 characters per line; the fancy version does the same but also includes a row of 64K RAM chips, so that your Apple contains 128K of RAM altogether.

What's included? The $749 price typically includes the Apple 2e and the fancy version of the 3A card, so that you get 80 characters per line and 128K. But the $749 price does *not* include a monitor or a disk drive or any cards in any other slots; they're "options". Moreover, each card costs about $100. To make matters worse, the $749 price doesn't even include any manuals: the manuals cost $50 extra. By the time you get done buying an Apple 2e plus a monitor, disk drive, second disk drive, disk controller card, printer card, and manuals, you'll have spent nearly $1400 if you bought from discount dealers, and even more if you paid retail prices.

Apple versus IBM clones. A complete Apple 2e system costs about the same as a Leading Edge Model D. In almost every way, the Apple 2e system is worse than the Leading Edge Model D: for example, the Apple 2e system has less RAM (128K instead of 256K), fewer keys on the keyboard (63 instead of 83), inferior disk drives (they write just 140K on the disk instead of 360K), and a crippled version of BASIC (it understands just 114 words instead of 178). Nevertheless, the Apple 2e outsells the Leading Edge Model D, because the Apple 2e lets you start cheaply (at just $749) and withhold later payments until your needs grow.

Also, more educational programs and games are available for the Apple 2e than for any other computer. That's because the Apple 2e still runs thousands of programs that were invented years ago for its predecessors: the Apple 2 and Apple 2 plus. Fewer educational programs and games have been written for the IBM PC and clones, because the IBM PC costs more than schools and kids can afford, and because the IBM PC is still too new to have developed a huge library of nifty software. Although the IBM PC has become the standard computer for business, the Apple 2e is still the standard computer for schools and for kids.



APPLE 2C

In 1984, Apple created a slight variation of the Apple 2e. The variation is called the *Apple 2 compact (Apple 2c)*. It's smaller than the Apple 2e and also weighs less. It usually sells for the same price as the Apple 2e: $749.

Which is better: the 2e or the 2c? On the one hand, the 2c has the convenience of being more portable; and it saves you money, because it consumes less electricity, and because its price includes a disk drive and two serial ports, which would cost you extra if you were buying a 2e. On the other hand, advanced users dislike the 2c because it's not expandable: it doesn't have any slots for adding cards. If you're a beginner, you won't want to add any cards anyway, since its motherboard already includes 128K of RAM, 80-character-per-line video circuitry, a disk controller, and two serial ports, and since you can run cables from the back of the 2c to a serial printer, modem, second disk drive, and joystick.

When the 2c first came out, its ROM was fancier than the 2e's, and gave the machine a better way to handle BASIC and a mouse. But in February 1985, Apple began putting the fancy ROM chips in the 2e also, so that every new 2e handles BASIC and a mouse as well as a 2c. (If your 2e is old, you can upgrade its ROM for $70.)

APPLE'S EXPENSIVE COMPUTERS

Apple Computer Inc. has invented ten computers. Five of them (the Apple 1, Apple 2, Apple 2-plus, Apple 2e, and Apple 2c) had mid-range prices. The other five (the *Apple 3*, *Lisa*, *Macintosh*, *Macintosh XL*, and *Macintosh Plus*) were expensive. Let's examine the expensive ones now. . . .

APPLE 3

The *Apple 3* resembled the Apple 2e but was much fancier and much more expensive. Because it was over-priced, and because the first ones off the assembly line were defective, and because it couldn't run some of the Apple 2-plus software, few people bought it.

When IBM finally invented the IBM PC, and consumers realized that the IBM PC was better and cheaper than the Apple 3, interest in the Apple 3 vanished, and Apple gave up trying to sell it.

LISA

The *Lisa* was even more expensive than the Apple 3, and it didn't run *any* Apple 2-plus software; in fact, it had a completely different CPU. But the Lisa received high praise, because its screen could draw fancy graphics quickly, and because its operating system and business programs were extremely easy to learn how to use: you just pointed to pictures, instead of typing hard-to-remember computer commands.

The Lisa's screen displayed cute little drawings, called *icons*. Some of the icons stood for activities. To make the Lisa perform an activity, you looked on the screen for the activity's icon. (For example, to make the computer delete a file, you looked for a picture of a garbage can.) When you pointed at the icon by using a mouse, the Lisa performed the icon's activity.

The Lisa also used *horizontal menus* and *pull-down menus*. A horizontal menu was a list of topics, printed across the top line of the screen. If you pointed at one of those general topics (by using the mouse), a column of more specific choices appeared under the general topic you pointed at; that column of specific activities was called a *pull-down menu*. You then looked at the pull-down menu, found the specific activity you were interested in, pointed at it (by using the mouse), and the computer would immediately start performing that activity.

Pointing to icons, horizontal menus, and pull-down menus was much easier to learn than using the kinds of computer systems other manufacturers had developed before. It was also fun! Yes, the Lisa was the first computer whose business programs were truly fun to run. And because it was so easy to learn how to use, customers didn't even have to read any manuals. Everybody praised the Lisa and called it a new breakthrough in software technology.

For the Lisa, Apple invented some business programs that were fun, cute, and easy to use. But independent programmers had difficulty developing their *own* programs for the Lisa, because Apple didn't supply enough programming tools: Apple never invented a version of BASIC, delayed introducing a version of PASCAL, and didn't make detailed manuals available to the average programmer. And although icons and pull-down menus are easy to use, they're difficult for programmers to invent.

Apple gradually lowered the Lisa's price.

MACINTOSH

In January 1984, Apple introduced the *Macintosh*, which was a stripped-down version of the Lisa. Like the Lisa, the Macintosh uses a mouse, icons, horizontal menus, and pull-down menus. The Macintosh consists of three parts: the mouse, the keyboard, and the system unit.

The system unit contains a 9-inch black-and-white screen, a 3½-inch floppy disk drive, and a motherboard. On the motherboard sit a 68000 CPU, two ROM chips (containing most of the operating system and many routines for drawing graphics), rows of RAM chips, a disk controller, and two serial ports (for attaching a printer and a modem).

Versions of the Macintosh. Apple began selling the Macintosh for $2495. The original version included 64K of ROM and 128K of RAM; it was called the *Slim Mac*. Then Apple invented an improvement, called the *Fat Mac*, which included 512K of RAM instead of 128K, because it used two rows of 256K chips instead of two rows of 64K chips.

Apple has stopped selling the Slim Mac. The Fat Mac lists for $1999, but many stores sell it for just $1549, and you can get it for under $1000 if you're a student or professor at a university that buys large quantities and gets an educational quantity discount.

Because the Mac is so easy to use and costs so little, many people have bought it, and lots of software has been developed for it—much more than for the Lisa. Unfortunately, the Lisa differs from the Macintosh enough so that the popular Macintosh software will not run on the Lisa well. Apple has stopped selling the Lisa, and has also stopped selling a compromise called the *Macintosh XL*.

In January 1986, Apple began selling a new, improved Macintosh, called the *Macintosh Plus*, which lists for $2599. It contains a larger ROM (128K instead of 64K), a larger RAM (1 megabyte instead of 512K), a better disk drive (double-sided instead of single-sided), a larger keyboard (which contains extra keys), and a port so that you can add a hard-disk drive more easily. The improved ROM, RAM, disk drive, keyboard, and port all serve the same overall purpose: they provide hardware and software tricks that let all your Mac programs run much faster.

What to add to your Mac. When you buy a Mac, you get a system master disk at no extra charge. Practically everybody who buys a Macintosh also buys a $195 disk that contains Apple's word-processing program (*Macwrite*) and also contains Apple's graphics program (*Macpaint*). To use Macwrite and Macpaint well, you'll need to buy Apple's *Imagewriter 2* printer, which uses an impact matrix and lists for $595.

Some dealers give you Macwrite and Macpaint free when you buy a Mac. They also give you a $125 discount on the Imagewriter 2, so that it costs you $470 instead of $595. Ask before you buy.

The Mac doesn't include any programming languages: they come on disks that cost extra. Get either *Microsoft BASIC* ($150) or *MacPASCAL* ($125). The current version of Microsoft BASIC is called *version 2.1*; do *not* use version 1, which is obsolete.

To do serious work on the Macintosh, you must buy a second disk drive, which can be either a single-sided floppy ($395), a double-sided floppy ($495) or a 20-megabyte hard ($1499). Those are the list prices, but most dealers charge less. For example, some dealers sell the double-sided floppy drive for $379 instead of $495.

So to get a complete decent Macintosh system, you begin by getting the $2599 Macintosh Plus (since the Fat Mac is too slow), then add the $195 Macwrite/Macpaint disk, the $595 Imagewriter 2 printer, the $150 Microsoft BASIC, and $495 second double-sided floppy drive. Altogether, the total list price is $4034. But since many dealers offer discounts, you can get the entire package for about $3500.

How to get discounts. To find discounts, ask mail-order dealers (such as Conroy-LaPointe), exhibitors at computer shows, and your neighborhood computer stores (when they hold special sales).

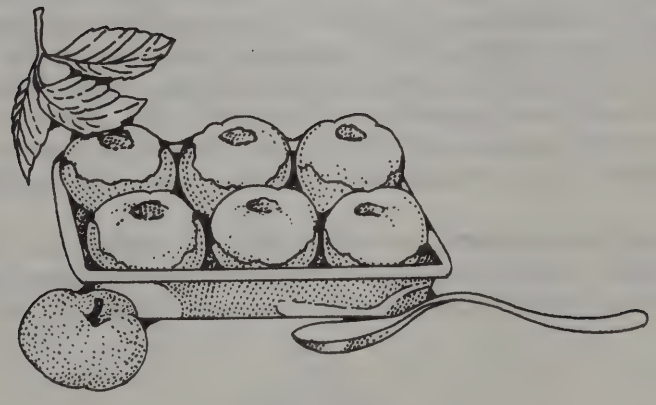
If you live near Boston, explore *Northeast Computer Stores*, which is a chain of stores that give members of the Boston Computer Society a 16% discount on Mac hardware. Since joining the society costs just $28, that's a good deal! You can reach the Boston Computer Society at 617-367-8080, and Northeast Computer Stores at 617-350-0032.

Mac versus IBM & clones. Even if you get a discount, a Mac will cost you more than a comparably equipped IBM PC clone (such as the Leading Edge Model D). Also, remember that the screen is small (just 9-inch) and can't handle colors yet, although Apple might create a color Mac someday.

Which is better, a Mac or an IBM PC? When the Mac was first invented, the answer was simple: the Mac's software was easier to learn how to use than the IBM PC's, but the Mac operated too slowly and not enough software was available. Recently, the IBM PC and Mac have both improved: IBM PC programmers, inspired by the Mac, have developed Mac-like easy-to-use software that runs on the IBM PC; and Mac developers, cowering under the power of the IBM PC marketplace, have developed faster hardware and more software.

So the IBM PC and Mac are beginning to resemble each other more and more. The question of which one to buy will ultimately become a question of personal taste.

The IBM PC appeals to people who are efficient, businesslike, conservative; the Mac appeals to people who are inspired, artistic, radical. Although I still recommend IBM PC clones, my recommendation might shift if the Mac continues to improve.



COMMODORE

COMMODORE 64

If you don't have enough money to buy an IBM PC or a PC Junior or even a clone, consider buying a *Commodore 64*. It lists for $199, but most dealers sell it for about $150.

Why so cheap? Why does the Commodore 64 cost so much less than an IBM PC? Here are several reasons. . . .

First, the advertised price does *not* include a disk drive. Commodore's standard disk drive is called the *Model 1541*. It lists for $269; most dealers charge about $199.

That disk drive costs so little because it's terrible. It puts very little information on the disk: just single-sided single-density). It's ridiculously slow, because the cable that runs from it to the computer contains just one wire for transmitting data, instead of several wires in parallel. The drive needs repairs frequently, because the drive's head frequently goes out of alignment.

If you're too cheap to buy the disk drive, you can buy Commodore's cassette tape recorder instead, which costs about $40. But since the tape recorder is even slower and even more awkward to use, and since the most popular software comes only on disk, I recommend that you keep saving your pennies until you can afford a disk drive.

Another reason why the Commodore 64 is so cheap is that it comes with just 64K of RAM, whereas most other computers include 128K or 256K or even more.

Also, the Commodore 64's advertised price does not include a monitor. Commodore's standard color monitor is called the *Model 1702*. Most dealers sell it for about $199. The image on that monitor's screen is slightly fuzzy, because the monitor is composite instead of RGB. The image is *not* sharp enough to display 80 characters per line clearly, though it's adequate for displaying 40 characters per line. If you're too cheap to buy the monitor, you can attach your home's TV instead, whose image is even fuzzier than the monitor. Since the monitor and TV are both fuzzy, most software for the Commodore 64 displays just 40 characters per line, whereas IBM PC software displays 80 characters per line instead.

Another problem with Commodore's video is than the M looks too much like an N, and the B looks too much like an 8.

The version of BASIC that comes with Commodore 64 isn't as fancy as the version that comes with the IBM PC. In fact, it's even worse than the version that comes with the Apple 2e & 2c. For example, it doesn't even include a command to let you draw a diagonal line across the screen. You can improve the BASIC by inserting a ROM cartridge called *Simon's BASIC*, but that cartridge costs about $40 extra. (It makes Commodore's BASIC surpass Apple's, though still not get quite as good as IBM's.)

The Commodore 64's printer port is non-standard: it works only with the strange printers manufactured by Commodore, unless you pay extra and get a special adapter.

The Commodore 64's keyboard includes just 66 keys, and its layout is archaic.

The Commodore 64's CPU is the 6502, which is slower than the IBM PC's CPU, and is even slower than the 65C02 CPU in the current Apple 2e & 2c.

Why is the Commodore 64 popular? In spite of all those limitations, the Commodore 64 is still popular, because it costs so little. Even after you add on the price of a disk drive and a monitor, the total is still less than the price of a PC Junior or an Apple 2e or 2c. By adding Simon's BASIC to the Commodore 64, you create a combo that's better for programming than the Apple 2e or 2c yet costs less.

Because so many folks own the Commodore 64, many programmers have created programs for it. Many of the popular Apple programs have been rewritten, so that you can buy Commodore 64 versions of the same programs. Since Commodore 64 owners tend to be penniless folks who can't spend lots of money for computers or software, programmers sometimes charge less for Commodore versions than for Apple versions of programs, even though the Commodore versions have all the features of the Apple versions. So by getting a Commodore 64, you'll pay less for software than if you buy an Apple.

On the other hand, *some* popular Apple programs have *not* been translated to the Commodore 64 yet. Before you buy a Commodore 64, check whether the programs you want have been translated yet.

The Commodore 64 contains a fancy music synthesizer chip, which lets the Commodore 64 produce a wide variety of musical tone qualities. When the Commodore 64 plays music, it sounds much better than an IBM PC or PC Junior or Apple 2e or 2c.

Is the Commodore 64 okay? The Commodore 64 is adequate for students. Because of its poor disk system, I recommend that you do *not* trust it to hold your business files; and because of its poor screen display, I recommend that you do not use it for serious word-processing projects.

If you buy a Commodore 64, you'll probably outgrow it after a year, and you'll wish you had paid the extra money needed to get an Apple or IBM clone instead. So although the Commodore 64 is okay as a short-term solution for students, I don't recommend it as a long-term solution for adults.

The Commodore 64's popularity has been decreasing, because Apple, IBM, and clones have all dropped their prices to the point where they cost only *slightly* more than getting a Commodore 64. Also, during the past few years many fancy programs have been invented for the Apple and IBM; those programs require more than 64K of RAM, and so they can't be translated easily to the Commodore 64.

VIC-20

The Commodore 64 is Commodore's improvement over an earlier computer, called the *Vic-20*. Because the Vic-20 is considered "obsolete", you can buy it very cheaply: most dealers sell it for just $69. It includes just 5K of RAM, which is just enough to let you practice writing short programs. It's adequate for children taking their first programming course, though not adequate for a follow-up course.

You can buy games for the Vic-20: the games come on tapes or in ROM cartridges.

BLACK COMPUTERS

After inventing the Vic-20 and the Commodore 64, Commodore invented two black computers: the *Commodore 16* and the *Commodore Plus 4*. They include excellent versions of BASIC (even better than if you add Simon's BASIC to the Commodore 64).

Unfortunately, the improved ROM and improved hardware make those black computers slightly incompatible with the Commodore 64. Disks containing Commodore 64 software must be slightly revised before they'll work on the black computers. But programmers never got around to making those revisions, and so you can't buy any decent software for those computers.

In an attempt to solve the software shortage, the Commodore Plus 4 comes with a gigantic ROM that includes a word-processing program and a spreadsheet program and a filing program. But those programs are so terrible that they're useless.

Without available decent software, those computers are useless (except to programmers), and so only a few programmers have bought them. Commodore has stopped making them, though some dealers still have a few left in stock.

You can get the Commodore Plus 4 very cheaply (just $79!) from a Minneapolis liquidator called *Comb* (800-328-0609). At that ridiculously low price, it's the most cost-effective computer for young programmers.

COMMODORE 128

The *Commodore 128* is wonderful and popular. It runs all the Commodore 64 software, but also includes a better version of BASIC, a better keyboard, better video, and many other little improvements.

It lists for $349, but most dealers sell it for $259, and you can get it for just $234 from Harmony Computers (New York 718-627-1000, out-of-state 800-441-1144).

To help you get the most from your Commodore 128, Commodore has invented for it a superior disk drive, called the *Model 1571*: get it from Harmony for just $223. Commodore has also invented for it a superior RGB monitor, called the *Model 1902*.

AMIGA

The *Amiga* is Commodore's newest and fanciest computer. It contains three special chips that let it produce fast animated graphics in beautiful shades of color. Like the Macintosh, it uses a mouse and pull-down menus, but the Amiga's graphics are far superior to the Macintosh's.

Since the Amiga is new, hardly any software is available for it yet. It's not compatible with the Commodore 64 or Macintosh or IBM PC, although Commodore has recently developed a $200 add-on that lets the Amiga run *some* IBM PC software slowly and crudely.

The Amiga is somewhat expensive. Commodore originally charged $1295 for the Amiga itself, plus $495 for the color monitor. Few folks could afford to spend $1295+$495 for a computer system that was pretty but couldn't run any popular business software. Recently, Commodore has been running a special sale, where you get the color monitor *free* with your Amiga. That sale makes the Amiga system more affordable. If you love color graphics, get an Amiga.

The Amiga reminds me of the *Mindset*, which was a beautiful graphics computer that everybody praised and hardly anybody bought. The company that produced the Mindset went bankrupt. I wonder whether Commodore will go bankrupt also. Even though Commodore is very large, it's verged on bankruptcy several times. At the moment, Commodore is still alive, but its future is uncertain.

When the Amiga first came out, its operating system was unreliable: if you filled up the Amiga's RAM, the Amiga went crazy. But since Commodore has fixed that problem, and since Commodore has been giving away the color monitor for free, and since several major companies are starting to write software for the Amiga, the Amiga still stands a chance of becoming a smashing success.

ALTERNATIVE COMPUTERS

ATARI

Atari's early computers were called the *400*, the *800*, the *600 XL*, the *800 XL*, and the *1200 XL*. They're all obsolete, and Atari has stopped selling them. They've been replaced by the *65 XE* and the *130 XE*, which run all the software invented for the earlier computers. The 65 XE includes "about 65 thousand bytes of RAM" (actually, 64K); the 130 XE includes "about 130 thousand bytes of RAM" (actually, 128K). Harmony sells the 65 XE for $84, and sells the 130 XE for $119.

After buying an XE, you'll also want to buy a disk drive. Harmony sells Atari's drive (Model 1050) for $129.

Atari versus Commodore. The 130 XE competes directly against the Commodore 64 (which is cheap) and the Commodore 128 (which is more expensive). Which computer is the best buy? Let's compare. . . .

Like the Commodore 64, the Atari 130 XE is cheap, displays no more than 40 characters per line, and comes with an inadequate version of BASIC. Like the Commodore 64, the Atari 130 XE lets you upgrade the BASIC by inserting a ROM cartridge: Atari sells that cartridge for about $40 and calls it *Microsoft BASIC 2*.

Like the Commodore 128, the Atari 130 XE includes 128K of RAM.

The Atari 130 XE comes with inadequate manuals. After buying the computer, you must then buy a book to understand how to program it.

In 1985, Atari began selling a new computer, called the *520 ST*. It uses a mouse, but costs much less than the other mousey computers (the Macintosh and the Amiga). Atari sells the computer—including a 68000 CPU, a 512K RAM, a double-sided 3½-inch floppy drive, disks containing the operating system, BASIC, LOGO, a word processor, and a data-management system, and a monitor—for just $799. That price includes a monochrome monitor; if you wanted color instead of monochrome, you pay $999 instead. Those are the list prices, but most dealers offer discounts. For example, Harmony sells the color version for just $784 instead of $999.

When the 520 ST first came out, those prices were about half as much as the Macintosh and Amiga, so that, by comparison, the Macintosh and Amiga looked overpriced. Apple had to lower the Macintosh's price, so that today you can buy a 512K Macintosh for about $1599; Commodore lowered the Amiga system's price, so that you can buy an Amiga with color monitor for just $1295; but notice that the Atari 520 ST is still the cheapest, at a discounted price of $784!

The ST's version of BASIC, though not by Microsoft, is very similar. (In some ways, it's slightly better; in other ways, it's slightly worse.)

Programmers have created a reasonable quantity of softwre for the ST. Today, more software is available for the ST than for the Amiga. That's because the ST had a head start: Commodore didn't ship the Amiga until several months after Atari had begun shipping the 520 ST, and the Amiga at that time was too overpriced and too unreliable for programmers to take seriously yet.

Neither the 520 ST nor the Amiga have as much software yet as the Macintosh. That's why the Macintosh is more popular, even though it costs more.

The ST and Amiga could both eventually get enough software to become smashing successes, though it's still too early to tell.

When Apple announced the Macintosh Plus, which contains 1 megabyte of RAM, Atari responded by announcing the *1040 ST*, which contains 1 megabyte also and costs slightly more than the 520 ST.

RADIO SHACK'S COLOR COMPUTERS

To compete against the Commodore 64, Radio Shack invented the *Color Computer*, which is nicknamed the *Coco*. Folks who love it call it the *Hot Coco*, and have started *Hot Coco Magazine*. But other computerists consider the Coco to be just a useless toy—a parody of a computer—and call it *Coco the Clown*.

Radio Shack began selling the Coco in 1980—the year before IBM began selling the PC. Microsoft, which invented the Coco's BASIC ROM, also invented the IBM PC's. The Coco's BASIC ROM is actually Microsoft's "rough draft" of the ROM that went into the IBM PC. So a Coco can best be described as "an IBM PC that's not quite right yet". If you study the Coco's BASIC, you'll see that its vocabulary for handling graphics and music are almost the same as the IBM PC's, but slightly more awkward. If you can't afford an IBM PC or IBM PC clone but nevertheless want to practice writing programs for the IBM PC, the Coco is the perfect computer for you.

Versions of the Coco. Radio Shack has sold several versions of the Coco. The original version was called the *Coco 1*; then came a variant called the *Micro Coco* and an improvement called the *Coco 2*. Today, Radio Shack sells just the Coco 2; the earlier versions are obsolete.

Today, Radio Shack sells just one version of the Coco 2: the current version has 64K of RAM and 16K of ROM and lists for $199.95. Discount dealers (such as Computer Plus) sell it for $165.

Warning: if you try to pay even less by getting a *used* Coco 2, make sure it has the 64K of RAM (not 16K) and make sure it has the 16K of ROM, which Radio Shack calls the *Extended BASIC ROM* (not just the 8K *Standard BASIC ROM*).

Should you buy a Coco? Although the Coco's BASIC resembles the IBM PC's very closely, the Coco isn't as popular as the Commodore 64, for several reasons: the Coco costs more; the Coco's disk drive, though far superior to Commodore's, is also more expensive ($299.95); the colors that the Coco puts on the screen are drab and disgusting ("faded green", "cough syrup red", and other losers); the music sounds like a sore throat trying to sing while gargling Listerine; and the screen displays just 16 lines of text (instead of 24), displays just 32 characters per line (instead of 40 or 80), and displays only capitals (unless you buy special software that adds lower-case letters).

The poor colors, poor sounds, and low quantity of characters on the screen add up to a computer that's boring. Although Radio Shack's BASIC is more competent that the versions of BASIC that come with the Commodore 64 and Atari 800 XL, most kids would rather *not* buy a Coco, because the Coco's unexciting screen and sounds make them snore.

Nevertheless, if you want to someday become an IBM PC programmer but can't afford an IBM PC or clone, the Coco's your best bet. And the Coco, its drive, its BASIC, and its DOS are all built more reliably than the Commodore 64's.

POCKET COMPUTERS

Radio Shack sells pocket computers. They fit in your pocket, run on batteries, and include LCD screens. They come in three versions.

The cheapest version, the *PC-4*, costs $59.95 and is manufactured for Radio Shack by Casio. The middle version, the *PC-3A*, costs $99.95 and is manufactured for Radio Shack by Sharp; it's better than the PC-4 in three ways: it displays more characters on the screen (24 characters instead of 12), includes a fancier version of BASIC, and includes more RAM (4K instead of 2K). The fanciest version, the *PC-5*, costs $119.95; it resembles the middle version (PC-3A) but contains a fancier ROM, which includes a filing program and also definitions of extra scientific functions.

You can attach each of Radio Shack's pocket computers to a cassette tape recorder and a thermal printer. The cassette tape recorder costs $49.95. The other goodies (the thermal printer plus an interface from the cassette tape recorder to the computer) cost a total of about $120.

Although all those pocket computers are cheap and cute, their tiny screens and tiny keyboards and tiny memories will frustrate you. For a similar price, you can buy a Commodore Vic-20, which you'll enjoy much more, after you attach it to your home's TV. Get a pocket computer only if you need to use a computer outdoors or in an environment where TV's aren't allowed.

LAP COMPUTERS

Radio Shack, NEC, and Epson all sell inexpensive lap computers.

The most popular is Radio Shack's *Model 100*, because it's the cheapest ($499) and the easiest to learn how to use.

It includes an LCD screen that displays eight 40-character lines, a pleasant keyboard, a modem (so that you can attach the computer to your telephone, after you buy a $19.95 cable), a 24K RAM, and a 32K ROM (which includes BASIC, a word-processing program, some filing programs, and a telecommunications program). Many reporters use it, for taking notes that can be easily phoned to the newspaper.

For an extra $119.95, you can expand the RAM to 32K; but if you do, the battery will run out faster. Most users stay with 24K.

Radio Shack also makes several fancier lap computers: the *Tandy 200* ($999) and the *Tandy 600* ($1599). They're too expensive to be popular.

CP/M COMPUTERS

The first popular microcomputer was the *Altair*. Its manufacturer, *Mits*, began selling it in January 1975. other companies (such as IBM and Apple) didn't make microcomputers until later.

The Altair is no longer manufactured, and its manufacturer no longer exists. But lots of software was written for it, and much of that software still works on computers that are made today.

In the original Altair, the CPU was an Intel 8080, and the DOS was called the *Control Program for Microprocessors (CP/M)*.

That DOS was invented by an Intel engineer: Gary Kildall. He quit Intel and started his own company, called *Digital Research (DR)*, whose main business was selling CP/M to computer manufacturers.

Some of the Intel 8080's inventors quit Intel and started their own company, *Zilog*. They invented the amazing *Zilog Z-80 CPU*, which could run all the Intel 8080 software (such as CP/M) and ran fancier software also.

The Z-80 quickly became the most popular CPU for microcomputers, while CP/M remained the most popular DOS—until IBM invented the IBM PC, which used an even fancier CPU (the Intel 8088) and an even fancier DOS (PC-DOS or MS-DOS).

After IBM began selling the IBM PC, hardly anybody wanted to buy the Z-80 or CP/M anymore. The companies who built Z-80 CP/M computers responded by lowering their prices, so now you can buy Z-80 CP/M computers very cheaply. But most Z-80 CP/M software is old-fashioned, boring (because it lacks graphics), and not as easy to use as the new IBM PC software.

KAYPRO

The most popular Z-80 CP/M computer is the *Kaypro 2X*, which lists for $1595. It includes 64K of RAM, two double-sided 5¼-inch floppy drives, a 9-inch green monitor, and a modem. The computer's transportable: it comes with a handle, so that you can carry the entire system in one hand.

The price also includes—at no extra charge—several disks full of software. The disks include CP/M, Microsoft BASIC, two other versions of BASIC (CBASIC and SBASIC), a word-processing program (Wordstar), a spelling-correction program (The Word Plus), two spreadsheet programs (Calcstar and Profit Plan), a data-management program (Infostar), and a telecommunications program (Mite). Unfortunately, most of that software's old-fashioned and awkward.

Kaypro also sells a souped-up version, called the *Kaypro 10*, which resembles the Kaypro 2X but includes extra data-management software (DBASE 2) and a 10-megabyte hard disk drive (instead of the second floppy drive). It lists for $2795.

Since CP/M's popularity is waning, Kaypro has recently stopped manufacturing the Kaypro 10 and is gradually phasing out the Kaypro 2X, so that Kaypro can devote more energy to building IBM PC clones instead.

RADIO SHACK'S Z-80 COMPUTERS

The *TRS-80 Model 1* was Radio Shack's first computer. Later, Radio Shack invented the Models 2, 3, 4, 4D, 4P, 12, 16, and 16B, and the Tandy 6000. Like the Model 1, they all contained a Z-80 CPU.

Most of those computers have become obsolete. The only ones that Radio Shack still makes are the Model 4D and the Tandy 6000.

Model 4D. The Model 4D includes a 64K RAM, 14K ROM, 12-inch green monitor, two double-sided 5¼-inch floppy drives, a disk containing Radio Shack's operating system (TRSDOS version 6) and Microsoft BASIC, and a disk containing an integrated package (Desk Mate). It lists for $1199, but discount dealers sell it for just $889.

Since it runs most of the software that was invented for its predecessors (Models 1, 3, 4, and 4P), it's bought by schools and companies that already own the predecessors and need more computers that are compatible. It can also run CP/M software, but most of the business and educational software developed for TRSDOS is newer and better than the software developed for CP/M.

Since software for the IBM PC is even newer and better than the software developed for TRSDOS, and since IBM PC clones are getting cheaper, interest in the Model 4D is fading.

Tandy 6000. The Tandy 6000 uses larger floppy disks (8-inch) and larger RAM (512K). It can handle three keyboards and screens simultaneously, so that a group of three people can share data and software.

It comes with *two* CPU's: a Z-80 (which handles old single-user software) and a 68000 (which handles new multi-user software). It uses an operating system called *Xenix* (pronounced "zeenicks"), which is Microsoft's version of the *Unix* operating system originally developed by Bell Labs.

In February 1986, Radio Shack dramatically slashed the Tandy 6000's price. The base price is now just $2499, for a single-user system. To handle three people well, you'll need to add two terminals, extra RAM, and a hard disk, bringing the grand total to about $6000.

COLECO'S ADAM

In 1983, Coleco began selling the *Adam* computer, which contained a Z-80 CPU. But instead of using CP/M or TRSDOS, the Adam used an operating system that imitated Apple DOS.

The Adam became very popular, because it was easy to learn how to use, its version of BASIC imitated Apple's, and its low $695 price included even an easy word-processing program and a daisy-wheel printer.

Unfortunately, the first Adams that came off the assembly line were defective. To make matters worse, instead of admitting that the Adams were defective, Coleco pretended the Adams worked perfectly: Coleco dishonestly told the press that there were "no returns", though actually 50% of the consumers who bought Adams were returning them for repairs.

The many Adam defects, combined with Coleco's dishonesty and Coleco's reluctance to distribute technical manuals about the Adam, discouraged programmers from developing software for the Adam.

In January 1984, Coleco quietly apologized. It began shipping improved Adams that worked perfectly. It even agreed to give the improved Adams free to customers who had bought the older ones and were complaining.

But Coleco's improvements came too late: software developers and journalists had lost faith in Coleco and switched their energies to other companies instead. Programmers, journalists, and consumers ignored the Adam. By the summer of 1984, hardly anybody was buying Adams anymore.

During the fall of 1984, Coleco dramatically lowered the Adam's price to $499. Moreover, Coleco promised to give a $500 college scholarship to any kid who bought an Adam, provided the kid was under 18 and could maintain good grades in college. So for those kids, Coleco was in effect giving away the Adam free!

Since Coleco said it would pay the $500 as $125 per year (for four years), and since Coleco wasn't making any profit by selling the Adams that way, industry analysts predicted that Coleco would go bankrupt before completing the four $125 payments.

Coleco hasn't gone bankrupt yet, because it's made lots of money by manufacturing Cabbage Patch dolls. Today you can still buy the Adam, for $299 or even less, from your local toy store. But since hardly any software is available for it, I don't recommend it.

Adam's word-processing program, though easy to learn how to use, works very slowly and is also very limited: for example, it doesn't contain any command to center a title, and it can't make the right margin perfectly straight. Though it's better than the typical electronic typewriter, it's not as powerful as the word-processing programs provided by most other computer manufacturers.

Adam's BASIC comes on a tape that cannot be copied easily and that "wears out" after about 3 months, so that you'll need to buy replacement tapes repeatedly: they cost about $30 each.

Because the Adam is cheap and easy to learn how to use (though inadequate), it appeals mainly to consumers who have very little money and are somewhat afraid of computers.

After playing with the Adam for six months, many of those consumers get excited enough about computers and disappointed enough in the Adam (because of the lack of software and the disintegrating BASIC tape) so that they stop using the Adam and buy a Macintosh instead—which costs six times as much—because the Macintosh, like the Adam, is advertised as being easy to use. But the Macintosh, unlike the Adam, now has lots of high-quality software.

Because Adam owners often buy Macintoshes later, the Adam has been called "the Macintosh's appetizer". In fact, if Apple were to buy Coleco and heavily advertise the Adam, Apple would ultimately sell more Macintoshes!

WOW! YOU FINISHED CHAPTER 3: BUYER'S GUIDE!
TURN THE PAGE FOR CHAPTER 4: PERSONAL PROGRAMMING.

ENTERING BASIC

CHOOSING A LANGUAGE

To operate a computer, you put your fingers on the computer's keyboard and type commands. You type the commands in English.

The computer understands only a small part of English; it understands just a few words and phrases. The words and phrases the computer understands are called the *computer's language*.

What's BASIC? Most computers use a language called *BASIC*, which was invented at Dartmouth College by two professors: John Kemeny and Tom Kurtz. They invented it in 1964 and later improved it. Today, BASIC consists of words such as PRINT, GO, TO, INPUT, IF, and THEN. This chapter explains how to use those words.

Microsoft BASIC. Different computers speak different *dialects* of BASIC. The most popular dialect was invented in 1975 by a 19-year-old kid named Bill Gates. Because he developed software for microcomputers, he called himself "Microsoft" and called his BASIC dialect "Microsoft BASIC".

Since Microsoft BASIC is so wonderful, all the popular computer companies paid him, to make their computers understand Microsoft BASIC. That's right: IBM, Commodore, Apple, Radio Shack, and *hundreds* of other computer companies all had to pay off Bill.

Microsoft BASIC has become so popular that Bill had to hire dozens of employees to help him fill all the orders. Microsoft Incorporated has become a multi-million-dollar company.

Today, Bill is rich, famous, handsome, 29 years old, and still single. Although he says he's too busy with computers to have time for marriage, maybe you or your friends can change his mind!

Over the years, Bill gradually improved Microsoft BASIC. Some computers use old versions of Microsoft BASIC; other computers use his latest improvements.

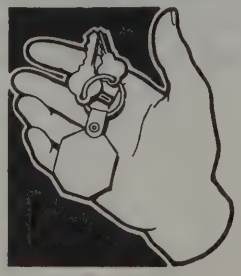
What's in this book? The Secret Guide to Computers explains all the major computer languages: BASIC, LOGO, PASCAL, C, DBASE, COBOL, and dozens of others. It begins by explaining *Modern Microsoft BASIC*, which includes the best features of Microsoft BASIC. All the popular microcomputers use either Modern Microsoft BASIC or a slight variation of it.

At the end of this volume, you'll find an amazing appendix, which reveals the intimate details of computers by IBM, Commodore, Apple, Radio Shack, Kaypro, Atari, and dozens of other manufacturers, including yours. It reveals the subtle ways in which *your* computer's BASIC differs from Modern Microsoft BASIC.

If you don't have access to a computer yet, ignore the appendix, and read just the main text, which explains Modern Microsoft BASIC.

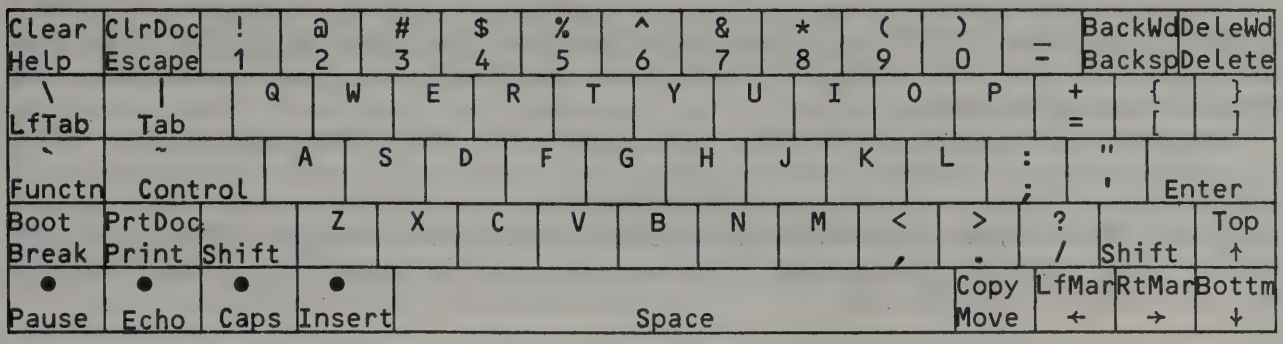
If you *do* have access to a computer, read the main text (which explains Modern Microsoft BASIC); but before trying any of the examples on your computer, peek at the appendix, to see how your computer differs from Modern Microsoft BASIC. To flip to the appendix easily, put a bookmark in the appendix. Do it *now*!

And remember: if you ever get confused about how to program *your* computer, phone me at 617-666-2666, and I'll help you, free!



THE KEYBOARD

Each year, manufacturers improve their keyboards. I predict that by 1990, most manufacturers will use the *Walter keyboard*, which I invented recently and looks like this:



Now I'm going to explain how to use the most important keys. But don't touch the keys on *your* keyboard, until you've read the next section (which is called GET STARTED) and until you've also read the appendix (which reveals how *your* keyboard differs from the Walter keyboard).

Numbers. Look at the Walter keyboard. The top row of keys contains the numbers.

Don't confuse 1 with I; the number 1 is in the top row. Don't confuse zero with the letter O; the number zero is in the top row.

(On older keyboards, the zero has a slash through it, like this: Ø. That's called a *slashed zero* or a *Swedish zero*.)

The SHIFT key. One of the keys looks like this: $\boxed{\begin{matrix}\$\\4\end{matrix}}$. If you press it, you'll type a 4. Here's how to type a dollar sign instead: hold down the SHIFT key; and while you keep holding down the SHIFT key, tap the $\boxed{\begin{matrix}\$\\4\end{matrix}}$ key. Here's the general rule: if a key has two symbols on it, and you want to type the top symbol, hold down the SHIFT key.

About the SHIFT key, beginners often make two boo-boos. The first boo-boo is forgetting to press it. For example, suppose you want to type a dollar sign, and you press the $\boxed{\begin{matrix}\$\\4\end{matrix}}$ key, but you forget to hold down the SHIFT key. Then you'll be typing a 4 instead.

The other boo-boo is trying to hit the SHIFT key and another key at exactly the same time. You can't do it; it's impossible; you'll wind up hitting one key before the other. The trick is to hold down the SHIFT key *first*; and while you keep holding it down, give the other key a light tap.

Beginners often make those boo-boos. When they realize their mistakes, they say "Oh, shit!" That's why programmers call it the "shit" key.

The keyboard contains *two* SHIFT keys. They serve the same purpose as each other. Use whichever SHIFT key makes your fingers feel more comfortable.

The CAPS key. If you tap the CAPS key, the computer will automatically capitalize all the letters you type. For example, if you tap the CAPS key and then press the M and E keys, the screen will show a capitalized ME. The computer will continue to capitalize all the letters, until you tell the computer to stop capitalizing.

To make the computer stop capitalizing, just tap the CAPS key again. After you've told the computer to stop capitalizing, pressing the M and E keys make the screen show a small me—unless you simultaneously hold down the SHIFT key.

The BACKSPACE key. The BACKSPACE key says BACKSP on it. If you make a typing mistake, you can erase the mistake, by pressing the BACKSPACE key, which makes the computer erase the last character you typed. To erase *two* characters, press the BACKSPACE key *twice*.

On older keyboards, the BACKSPACE key has a long left-arrow on it, like this:

$\boxed{\leftarrow}$

The ENTER key. The most important key is the ENTER key. When using BASIC, press the ENTER key at the end of every line you type. The ENTER key takes the line you've typed and enters that line into the computer, so that the computer reads the line.

The computer ignores what you type, until you press the ENTER key. If you forget to press the ENTER key at the end of the line, the computer doesn't read what you typed, and so the computer doesn't do anything, and you wonder why the computer seems broken.

(On older keyboards, the ENTER key is called the RETURN key.)

GET STARTED

Make sure the various parts of your computer system are correctly attached to each other. Then turn on the power. If your computer is modern (and uses Modern Microsoft BASIC), it will say:

```
OK
```

(Older computers don't say OK. Instead, they say READY or print a bracket. For further details about *your* computer, read the appendix.)

Be bold. In science fiction, computers blow up; in real life, they never do. No matter what keys you press, no matter what commands you type, you won't hurt the computer. The computer is invincible! So go ahead and experiment. If it doesn't like what you type, it will gripe at you, but so what?

Troubles. When you try using the computer, you'll have trouble. It might be because you're making a mistake. It might be because the computer is broken. Or it might be because your computer is weird and works differently from the majority of computers discussed in this book. (Each computer has its own "personality", its own quirks.)

Whenever you have trouble, laugh about it, and say, "Oh, boy! Here we go again!" (If you're Jewish, you can say all that more briefly, in one word: "Oy!") Then get some help.

Get help. For further help with *your* computer, read the appendix. For even more help, read the beginner's manual that came with your computer, or ask the genie who gave you the computer (your salesperson or parent or boss or teacher or friend).

If you're sitting near computers in your office, school, or home, and other people are nearby, ask them for help. They'll gladly answer your questions, because they like to show off, and because the way *they* got to know the answers was by asking.

Computer folks like to explain computers, just as priests like to explain religion. Remember: you're joining a cult! Even if you don't truly believe in "the power and glory of computers", at least you'll have a few moments of weird fun. So play along with the weird computer people, boost their egos, and they'll help you get through your initiation rite. Above all, assert yourself, and *ask questions*. "Shy guys finish last."

When dealing with the computer and the people who surround it, be friendly but also aggressive. To make sure you get your money's worth from a computer course, ask your teacher and coworkers questions, questions, questions! If you're using a computer that you own, get help from the person who gave it to you.

Your town probably has a *computer club*. (To find out, ask the local schools and computer stores.) Join the club, and tell the club you'd like help with your computer. Probably some computer hobbyist will help you.

And remember—you can call *me* anytime at 617-666-2666, and I'll help you, free!

Let's try to make the computer print the answer to 5+2. (And let's hope the computer says 7.) Type this:

```
PRINT 5+2
```

Type that carefully! Here's how. First, type the word PRINT, by pressing the P key, then the R key, then the I key, then the N key, then the T key. (Since most computers don't care about capitalization, do *not* bother pressing the SHIFT key.) After typing the word PRINT, type a blank space (by pressing the SPACE bar). Then press the 5 key. Then type a plus sign, but be careful: on most keyboards, the plus sign is on the *top* part of a key; so to type the plus sign, you must hold down the SHIFT key. Then type the 2. When you've finished typing that line, press the ENTER key, which makes the computer read what you've typed.

Then the computer will print the answer:

```
 7
```

If your computer prints a wrong answer instead or says ERROR, you made a typing mistake. Find your mistake (by looking at the screen), and then try again to type

```
PRINT 5+2
```

Underneath the answer, the computer will print the word OK (or some old-fashioned word, such as READY). That means it's okay for you to feed the computer the next problem; the computer is ready for another.

If you want to subtract 3 from 7, type this:

```
PRINT 7-3
```

(To type the minus sign, you do *not* press the SHIFT key.) When you've finished typing that line, remember to press the ENTER key, which makes the computer read the line and print the answer. The computer will print:

```
 4
```

You can use decimal points and negative numbers. For example, if you type this—

```
PRINT -26.3+1
```

the computer will print:

```
-25.3
```

To multiply, use an asterisk. To multiply 2 by 6, type this:

```
PRINT 2*6
```

The computer will print:

```
 12
```

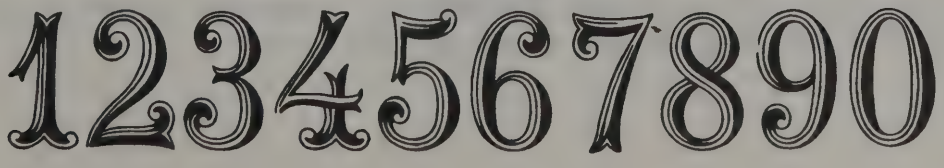
To divide, use a slash. So to divide 8 by 4, type this:

```
PRINT 8/4
```

The computer will print:

```
 2
```

Do *not* put commas in large numbers. To write four million, do *not* write 4,000,000; instead, write 4000000.



QUESTION MARK

If you're too lazy to type the word PRINT, type a question mark instead. For example, instead of typing—

```
PRINT 5+2
```

you can type:

```
? 5+2
```

The question mark is an abbreviation: it stands for the word PRINT.

If you prefer, you can think of the question mark as standing for the word "What's". For example, if you want to ask the computer "What's 5+2", you type:

```
? 5+2
```

HUGE & TINY NUMBERS

If the computer's answer is huge (more than a million) or tiny (less than .01), the computer might print the answer strangely. Here's how. . . .

E notation. The computer might print the letter E. The E means: move the decimal point.

For example, suppose the computer says the answer to a problem is:

```
 8.51673E+12
```

The E means, "move the decimal point". The plus sign means, "towards the right". Altogether, the E+12 means, "move the decimal point towards the right, 12 places." So look at 8.51673, and move the decimal point towards the right, 12 places; you get 8516730000000. So when the computer says the answer is 8.51673E+12, the computer really means the answer is 8516730000000, approximately. (The exact answer might be 8516730000000.2, or 8516730000001.79, or some similar number; what the computer tells you is just an approximation.)

Suppose your computer says the answer to a problem is:

```
 9.26E-04
```

After the E, the minus sign means, "towards the *left*". So look at 9.26, and move the decimal point towards the left, 4 places; you get:

.000926

So when the computer says the answer is 9.26E-04, the computer really means the answer is:

.000926

You won't have to worry much about E notation: the computer uses it only if an answer is huge (many millions) or tiny (tinier than .01). But when the computer *does* use E notation, remember to move the decimal point!

The highest number. The highest number the computer can handle is about 1E38, which is 1 followed by 38 zeros, like this:

10000000000000000000000000000000000000

If you try to go much higher, the computer will say:

```
OVERFLOW ERROR
```

The tiniest decimal. The tiniest decimal the computer can handle is about 1E-38, which is a decimal point followed by 38 digits, 37 of which are zeros, like this:

.00000000000000000000000000000000000001

If you try to go tinier, the computer will have an "underflow error" and will "fake" the answer: it will give zero instead of the correct answer.

ORDER OF OPERATIONS

What does "2 plus 3 times 4" mean? The answer depends on who you ask.

To a clerk, it means "start with 2 plus 3, then multiply by 4"; that makes 5 times 4, which is 20. But to a scientist, "2 plus 3 times 4" means something different: it means "2 plus three fours", which is 2+4+4+4, which is 14.

Since computers were invented by scientists, computers think like scientists. If you type—

```
PRINT 2+3*4
```

the computer will think you mean "2 plus three fours", so it will do 2+4+4+4 and print this answer:

```
 14
```

The computer will *not* print the clerk's answer, which is 20. So if you're a clerk, tough luck!

Scientists and computers follow this rule: do multiplication and division before addition and subtraction. So if you type—

```
PRINT 2+3*4
```

the computer begins by hunting for multiplication and division. When it finds the multiplication sign between the 3 and the 4, it multiplies 3 by 4 and gets 12, like this:

```
PRINT 2+3*4
        12
```

So the problem becomes 2+12, which is 14, which the computer prints.

For another example, suppose you type:

```
PRINT 10-2*3+72/9*5
```

The computer begins by doing all the multiplications and divisions. So it does 2*3 (which is 6) and does 72/9*5 (which is 8*5, which is 40), like this:

```
PRINT 10-2*3+72/9*5
          6   40
```

So the problem becomes 10-6+40, which is 44, which is the answer the computer prints.

Parentheses. You can use parentheses the same way as in algebra. For example, if you type—

```
PRINT 5-(1+1)
```

the computer will compute 5-2 and print:

```
 3
```

You can put parentheses inside parentheses. If you type—

```
PRINT 10-(5-(1+1))
```

the computer will compute 10-(5-2), which is 10-3, and will print:

```
 7
```

Let's make the computer fall in love. Let's make it say, "I love you".

Type this:

```
PRINT "I LOVE YOU"
```

Type it carefully. Begin by typing the word PRINT. (If you're lazy, type a question mark instead.) Then press the SPACE bar. Then type a quotation mark, but be careful: on most keyboards, the quotation mark is on the *top* part of a key; so to type the quotation mark, you must hold down the SHIFT key. Then type I LOVE YOU. Then type another quotation mark.

At the end of the line, press the ENTER key. The computer will obey your command; it will print:

```
I LOVE YOU
```

You can change the computer's personality. For example, if you give this command—

```
PRINT "I HATE YOU"
```

the computer will reply:

```
I HATE YOU
```

Notice that to make the computer print a message, you must put the message between quotation marks. The quotation marks make the computer copy the message without worrying about what the message means. For example, if you misspell "I love you", and type—

```
PRINT "AIEEE LUF YA"
```

the computer will still copy the message (without worrying about what it means); the computer will print:

```
AIEEE LUF YA
```

The word "JOY" consists of 3 characters: J and O and Y. Programmers say that the word "JOY" is a *string* of 3 characters.

A *string* is any collection of characters, such as "JOY" or "I LOVE YOU" or "AIEEE LUF YA" or "76 TROMBONES" or "GO AWAY!!!" or "XYPW EXR///746". The computer will print whatever string you wish, but remember to put the string in quotation marks.

Strings versus numbers. The computer can handle two types of expressions: *strings* and *numbers*. Put strings (such as "JOY" and "I LOVE YOU") in quotation marks. Numbers (such as 2+2) do *not* go in quotation marks.

What happens if you make a boo-boo? What happens if you accidentally put a number in quotation marks? For example, what happens if you accidentally put 2+2 in quotation marks, like this:

```
PRINT "2+2"
```

The quotation marks make the computer think "2+2" is a string instead of a number. Since the computer thinks "2+2" is a string, it copies the string, without analyzing what it means; the computer will print:

```
2+2
```

It will *not* print 4.

What happens if you make the opposite kind of boo-boo? What happens if you mean to type a string, but forget the quotation marks? For example, what happens if you mean to type "I LOVE YOU" but forget the quotation marks, and accidentally type this instead:

```
PRINT I LOVE YOU
```

Because you forgot the quotation marks, the computer thinks the I LOVE YOU is a weird number instead of a string; but the computer doesn't know what kind of weird number it is, since the computer doesn't know the meaning of love. Whenever the computer is confused, it either gripes at you or prints a zero. In this particular example, it will print a zero, like this:

```
 0
```

So if you incorrectly tell the computer to say it loves you, it will say zero.

LONG COMMANDS ON SMALL SCREENS

Try typing this command:

```
PRINT "I LIKE TO EAT AND DRINK AND GAZE INTO YOUR EYES AS WE DREAM ABOUT LOVE"
```

That command's long. It won't fit on a single line of your screen, if your screen's small; but go ahead and type the command anyway. When you reach the screen's right edge, the computer automatically moves you to the line below, so you can continue typing. Do *not* press the ENTER key, until you've typed the entire command.

After typing the entire command, press the ENTER key. The computer will print:

```
I LIKE TO EAT AND DRINK AND GAZE INTO YOUR EYES AS WE DREAM ABOUT LOVE
```

EASY PROGRAMS

HOW TO PROGRAM

Now that you've had fun experimenting with the word PRINT, you're ready for the big-time; you're ready to program the computer! Programming the computer consists of three steps. . . .

Step 1: type the word NEW. That tells the computer you're going to invent a new program. After you type the word NEW, press the ENTER key. The computer will say OK. (If your computer is old-fashioned, it will say some other word instead, such as READY.)

Step 2: type your program. A *program* is a list of numbered commands. For example, suppose you want the computer to say:

```
I LOVE YOU
YOU TURNED ME ON
LET'S GET MARRIED
```

Type this program:

```
1 PRINT "I LOVE YOU"
2 PRINT "YOU TURNED ME ON"
3 PRINT "LET'S GET MARRIED"
```

Every line of the program must be numbered; you must type those numbers. Every line of the program must say PRINT; you must type the word PRINT. (Or, if you're too lazy to type the word PRINT, you can type a question mark instead.) Every line of that program must have quotation marks; you must type the quotation marks. At the end of each line, press the ENTER key.

Step 3: type the word RUN. That tells the computer to look at the program you've written, and run through it. After you type the word RUN, press the ENTER key. The computer will print everything you requested; it will print:

```
I LOVE YOU
YOU TURNED ME ON
LET'S GET MARRIED
```

Then the computer will say OK.

Review. Let's review that procedure. You type:

```
NEW
1 PRINT "I LOVE YOU"
2 PRINT "YOU TURNED ME ON"
3 PRINT "LET'S GET MARRIED"
RUN
```

The numbered instructions (1, 2, and 3) are called the *program*. Above the program, you type the word NEW; below the program, you type the word RUN. The words NEW and RUN are *not* numbered. After you've done all that correctly (and pressed the ENTER key after the word RUN), the computer will obey the program and print what you requested.

Each line of the program must be numbered. Putting a number at the beginning of a line makes the line become part of your program; the computer won't obey the line until the program is RUN. Here's the rule: a number in front of the line tells the computer to wait until you type the word RUN.

Another example. For another example, try typing this:

```
NEW
1 PRINT "I LONG"
2 PRINT 2+2
3 PRINT "U"
RUN
```

At the end of each line, remember to press the ENTER key. When you press the ENTER key after the word RUN, the computer obeys the program. The first line makes the computer print "I LONG"; the second line makes the computer add 2 and 2, so the computer prints the number 4; the third line makes the computer print the letter U. Altogether, the computer prints:

```
I LONG
 4
U
```

Yes, the computer says it longs for you!

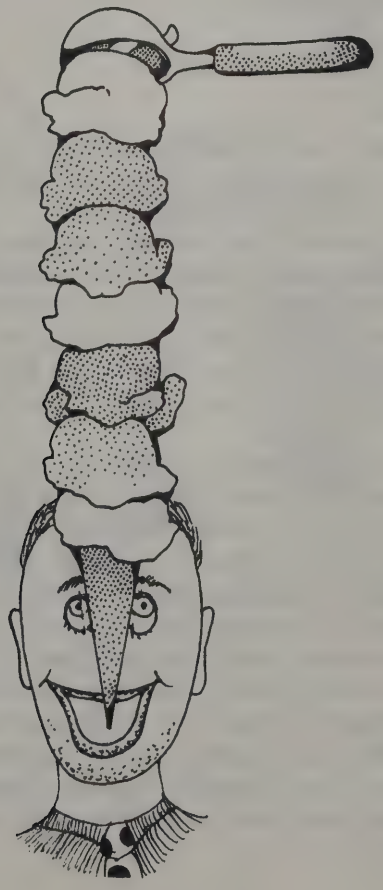
In that example, the computer indents the 4, so that the 4 has a blank space in front of it. That's because of this rule: the computer automatically puts a blank space in front of every positive number.

After the computer has finished running that program, you can type the word RUN again; the computer will print again:

```
I LONG
 4
U
```

A program is a list of numbered instructions. If the program you typed isn't on the screen anymore, and you'd like to see that list of numbered instructions again, type the word LIST. The computer will say:

```
1 PRINT "I LONG"
2 PRINT 2+2
3 PRINT "U"
```



Poem. Suppose you want to make multiple copies of this poem:

```
SPRING HAS SPRUNG.
THE GRASS HAS RIZ.
I WONDER WHERE
THE FLOWERS IS.
```

First, turn the poem into a program, by typing:

```
NEW
1 PRINT "SPRING HAS SPRUNG."
2 PRINT "THE GRASS HAS RIZ."
3 PRINT "I WONDER WHERE"
4 PRINT "THE FLOWERS IS."
```

Then type the word RUN (and press the ENTER key). The computer will print the poem. If you type the word RUN again, the computer will print the poem again. Each time you type the word RUN, the computer will print the poem. So to print many copies of the poem, just type the word RUN many times, like this:

```
You type→→→→→→→→→→→→RUN

The computer types→→SPRING HAS SPRUNG.
                    THE GRASS HAS RIZ.
                    I WONDER WHERE
                    THE FLOWERS IS.
                    OK

You type→→→→→→→→→→→→RUN

The computer types→→SPRING HAS SPRUNG.
                    THE GRASS HAS RIZ.
                    I WONDER WHERE
                    THE FLOWERS IS.
                    OK

You type→→→→→→→→→→→→RUN

The computer types→→SPRING HAS SPRUNG.
                    etc.
```

If you hate poetry and all other forms of art, cheer up: by using the same technique, you can make the computer type many copies of your favorite flubbed-up business letter, or your favorite obnoxious ad.

Become an expert. To invent your *own* program and make the computer handle it, go through the three steps again: type the word NEW, then type your program, then type the word RUN. Remember to press the ENTER key at the end of each line, and remember that the commands NEW and RUN are *not* numbered.

Those three steps (NEW then program then RUN) are all you have to know about programming! C'mon, write some programs! It's easy! Try it. You'll have lots of fun!

A person who writes a program is called a *programmer*. Congratulations: *you're* a programmer! Write several programs like the ones I've shown you already. Then you can put on your résumé that you have "a wide variety of programming experience", and you can talk your way into a programming job!

The rest of this chapter explains how to become a *good* programmer.

Programming a computer is like driving a car: the only way to become an expert is to put your hands on that mean machine and try it yourself.

If you have access to a computer, put this book next to the computer's keyboard. At the end of each paragraph, type the examples and look, look, see the computer run. Invent your own variations: try typing different numbers and strings. Invent your own programs: make the computer print your name or a poem; make it solve problems from your other courses and the rest of your life. The computer's a fantastic toy. Play with it.

If you're a student, don't wait for your instructor to give lectures and assign homework. *Act now*. You'll learn more from handling the computer than from the lectures or readings. Experience counts.

Let me tell you the story of Charlie. . . .

At Wesleyan University's computer center, one of the directors was having trouble making the computer print the university's payroll. He asked me for help, but I said I didn't know either. I saw a little kid sitting at one of the keyboards. "Hey, Charlie," I called to him, "we're having trouble getting the payroll out."

Little Charlie came over and typed some stuff on our keyboard. "The payroll will be out in a minute," he said gleefully.

Charlie was only in seventh grade. He'd never taken a computer course, because his school didn't offer one. But he'd spent the whole summer just "hanging around" our computer, and knew it better than we.

Be like Charlie. Hang around your computer. Communicate with it every day. At first, that will be even harder than talking with a cat or a tree, because the computer belongs to a different specie, a different kingdom; but keep trying. Get to know it as well as you know your best friend.

When dealing with the computer and the people who surround it, be friendly but also aggressive. To make sure you get your money's worth from a computer course, ask your teacher, classmates, lab assistants, and other programmers questions, questions, questions!

If you're taking a French course, you might find French difficult; and if you're taking a computer course, you might find computers difficult also. But even a stupid three-year-old French kid can speak French, and even kindergarten kids can program the computer. They've got only one advantage over you: practice!

MAKE A MISTAKE?

If you make a mistake, you can correct it in three ways. . . .

- If you notice the mistake before you press the ENTER key, correct the mistake by using the BACKSPACE key.
- If you mess up the whole program, rewrite it, by typing the word NEW and then starting from the beginning again.
- If you mess up a line, retype it underneath. For example, suppose you type—

```
1 PRIMT "I LOVE YOU"
```

and then you notice that you misspelled the word PRINT. Just retype the line, so your screen looks like this:

```
1 PRIMT "I LOVE YOU"
1 PRINT "I LOVE YOU"
```

For another example, suppose you type this tongue-twister—

```
1 PRINT "SHE SELLS"
2 PRINT "SEE SHELLS"
3 PRINT "BY THE SEASHORE"
```

and then you notice that in line 2, the SEE ought to be SEA. Underneath the program, retype that line, so your screen looks like this:

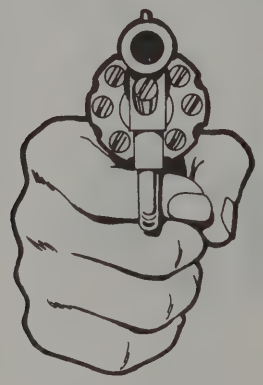
```
1 PRINT "SHE SELLS"
2 PRINT "SEE SHELLS"
3 PRINT "BY THE SEASHORE"
2 PRINT "SEA SHELLS"
```

When you type RUN, the computer will run the corrected version and print:

```
SHE SELLS
SEA SHELLS
BY THE SEASHORE
```

When you type LIST, the computer will list the corrected program; the computer will say:

```
1 PRINT "SHE SELLS"
2 PRINT "SEA SHELLS"
3 PRINT "BY THE SEASHORE"
```



What if the computer gripes? If the computer ever gripes at you, don't worry: just correct your mistake. For example, suppose you type—

```
1 PRIMT "ROSES ARE RED"
2 PRINT "CABBAGE IS GREEN"
3 PRINT "MY FACE IS FUNNY"
4 PRINT "BUT YOURS IS A SCREAM"
```

and you don't notice that you misspelled PRINT in line 1. If you type RUN, the computer will try to run the program but will be confused by line 1, so it will gripe at you. It will say:

```
SYNTAX ERROR IN 1
```

That means the computer hasn't the faintest idea of what you're talking about in line 1. Your next step is to correct the error. To do that, just retype line 1 correctly, and then type the word RUN again. So altogether, the conversation looks like this:

You type this, but it contains an error→
```
1 PRIMT "ROSES ARE RED"
2 PRINT "CABBAGE IS GREEN"
3 PRINT "MY FACE IS FUNNY"
4 PRINT "BUT YOURS IS A SCREAM"
RUN
```

The computer gripes at you→
```
SYNTAX ERROR IN 1
```

You type this, to correct the error→
```
1 PRINT "ROSES ARE RED"
RUN
```

The computer recites the poem→
```
ROSES ARE RED
CABBAGE IS GREEN
MY FACE IS FUNNY
BUT YOURS IS A SCREAM
```

Common bloopers. If you're like most beginners, you'll make these mistakes soon—if you haven't made them already!

You type the letter O instead of zero, or type zero instead of the letter O. Your typing looks correct to you, but the computer gripes.

You type a command (such as RUN) but you forget to press the ENTER key afterwards. The computer keeps waiting for you to press it. Since you don't realize the computer's waiting for you, and since the computer isn't replying, you think the computer is broken.

You tell the computer to PRINT a message, but you forget to put the message in quotation marks. So the computer gripes or prints a zero instead.

You start typing a new program, but forget to type the word NEW. So when you type RUN, the computer runs a mish-mash of the new program with the previous program, and reprints some messages from the previous program.

HOW THE COMPUTER REARRANGES YOUR PROGRAM

You don't have to type your program in order. For example, suppose you type:

```
4 PRINT "TOES"
2 PRINT "TOAD"
3 PRINT "LICKS"
1 PRINT "MY"
```

In its mind, the computer automatically rearranges the program, so the numbers are in increasing order, like this:

```
1 PRINT "MY"
2 PRINT "TOAD"
3 PRINT "LICKS"
4 PRINT "TOES"
```

If you'd like to peek inside the computer's mind and see what program the computer is thinking of, type the word LIST, like this:

You type this program→→→

If you type RUN, the computer will print:

```
MY
SORE FEET
NEED A
TOE
TRUCK
```

If you think about that example, you'll notice two things:

To revise a line, just retype it.

To erase line 80, just type—
80
with nothing else on the line.

The word NEW erases your program from the computer's mind, to make way for a new one. But the word RUN does *not* erase the program; after the run, you can continue inserting, deleting, and revising lines.

NUMBERING BY TENS

If your first line is numbered 1, and your second line is numbered 2, you can't squeeze a line between them, since decimals aren't allowed. So expert programmers number their lines 10, 20, 30, 40, . . . instead of 1, 2, 3, . . .

That way, if your first line is numbered 10 and your second line is numbered 20, and you want to insert an extra line between them, you can call it 15. If you later want to insert an extra line between 10 and 15, you can call it 12. If you want to insert another line between 10 and 12, you can call it 11.

RANGES OF LINES

If your program's too long to fit on your screen, tell the computer to list *part* of your program. For example, to list just lines 30 through 80, type this:

```
LIST 30-80
```

To list from line 30 to the end of your program, type this:

```
LIST 30-
```

To list from the beginning of your program up to line 80, type this:

```
LIST -80
```

To list just line 30, type this:

```
LIST 30
```

To delete lines 30 through 80, type this:

```
DELETE 30-80
```

That works on most computers. Some computers are different; check the appendix.

COLONS

You can put several statements on the same line, like this:

```
10 PRINT "I THINK": PRINT "OF FLEAS": PRINT "AND SNEEZE"
```

The computer will PRINT "I THINK". Then the computer will PRINT "OF FLEAS". Then the computer will PRINT "AND SNEEZE". Altogether, the computer will print:

```
I THINK
OF FLEAS
AND SNEEZE
```

TRICKY PRINTING

SEMICOLONS

Run this program:

```
10 PRINT "FAT";
20 PRINT "HER"
```

Line 10 makes the computer print FAT; and line 10 ends with a semicolon. The semicolon makes the computer print the next item on the same line; so the computer will print HER on the same line, like this:

```
FATHER
```

This command shows what happens to an evil king on a boat:

```
PRINT "SIN";"KING"
```

The computer will print SIN, and will print KING on the same line, like this:

```
SINKING
```

SPACES

Sometimes life consists of joy; sometimes life consists of sorrow; but sometimes life consists merely of a numb, empty feeling that's neither joy nor sorrow. To express that feeling, run this program:

```
10 PRINT "JOY"
20 PRINT
30 PRINT "SORROW"
```

Line 10 makes the computer print JOY. Line 20 makes the computer print a blank, empty line, underneath JOY. Altogether, the computer will print:

```
JOY

SORROW
```

Suppose you want the computer to print this letter:

```
DEAR JOAN,
  THANK YOU FOR THE BEAUTIFUL
NECKTIE.  JUST ONE PROBLEM--
I DON'T WEAR NECKTIES!
              LOVE,
              FRED-THE-HIPPIE
```

This program prints it:

```
10 PRINT "DEAR JOAN,"
20 PRINT "  THANK YOU FOR THE BEAUTIFUL"
30 PRINT "NECKTIE.  JUST ONE PROBLEM--"
40 PRINT "I DON'T WEAR NECKTIES!"
50 PRINT "              LOVE,"
60 PRINT "              FRED-THE-HIPPIE"
```

In the program, each line contains two quotation marks. To make the computer indent a line, put blank spaces *after* the first quotation mark.

Spaces after numbers. Try typing this command:

```
PRINT -3;"IS MY FAVORITE NUMBER"
```

Whenever the computer prints a number, it prints a blank space afterwards; so the computer will print a blank space after the -3, like this:

```
-3 IS MY FAVORITE NUMBER
```

space

Spaces before numbers. This command tells what to put in your coffee:

```
PRINT 7;"DO";"NUTS"
```

The computer prints 7 and DO and NUTS. Since 7 is a number, the computer prints a blank space after the 7. The computer prints another blank space *before* every number that's positive; so the computer prints another blank space before the 7, like this:

```
 7 DONUTS
```

spaces

Hey, if you're feeling cool, maybe this command expresses your feelings:

```
PRINT "THE TEMPERATURE IS";4+25;"DEGREES"
```

The computer prints THE TEMPERATURE IS, then 4+25 (which is 29), then DEGREES. Since 29 is a positive number, the computer prints a blank space before and after the 29:

```
THE TEMPERATURE IS 29 DEGREES
```

spaces

Use this command if you're even colder:

```
PRINT "THE TEMPERATURE IS";4-25;"DEGREES"
```

The computer prints THE TEMPERATURE IS, then 4-25 (which is -21), then DEGREES. Since -21 is a number, the computer prints a space after it; but since -21 is *not* positive, the computer does *not* print a space before it. The computer prints:

```
THE TEMPERATURE IS-21 DEGREES
```

no space space

Yuk! That looks ugly! It would look prettier if there were a space before the -21. To insert a space, put the space inside quotation marks:

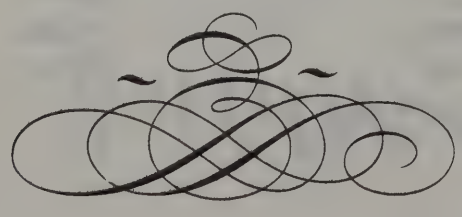
```
PRINT "THE TEMPERATURE IS ";4-25;"DEGREES"
```

inserted space, before the quotation mark

Then the computer will print:

```
THE TEMPERATURE IS -21 DEGREES
```

inserted space



MULTIPLE CALCULATIONS

By using semicolons, you can make the computer do many calculations at once. For example, suppose you want the computer to do 6+2, and also 6-2, and also 6*2, and also 6/2. This command makes the computer do all those calculations at once:

```
PRINT 6+2;6-2;6*2;6/2
```

The computer prints the four answers:

```
 8  4  12  3
```

The computer prints spaces between the answers, because the computer prints a space after every number (and an additional space before every number that's positive).

CLEARING THE SCREEN

The Walter keyboard includes a CLEAR key. If you press the CLEAR key, the computer erases the entire screen, so that the screen becomes cleared and empty.

Though pressing the CLEAR key erases the screen, it does *not* erase the computer's memory; the computer still remembers what program you were working on. The only way to erase the computer's memory is to type the word NEW.

So whenever you want to begin a new program, type the word NEW, which erases the previous program; pressing the CLEAR key is *not* sufficient.

You can begin your program with this line:

```
10 CLS
```

It makes the computer automatically CLear the Screen, before doing the rest of the program.

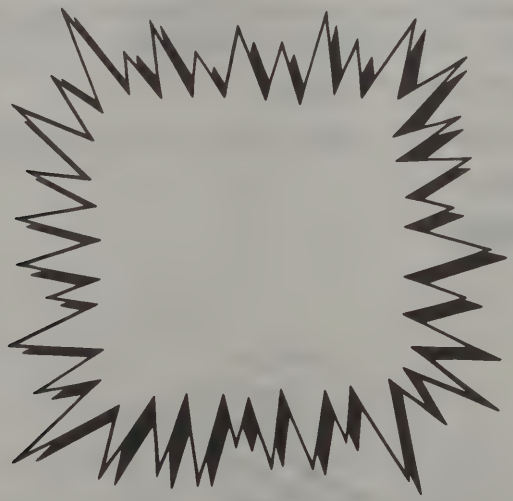
For example, run this program:

```
10 CLS
20 PRINT "I'M SITTING ON TOP OF THE WORLD"
```

Line 10 makes the computer CLear the Screen. Since the screen's been cleared, line 20 prints this message *at the top of the screen*:

```
I'M SITTING ON TOP OF THE WORLD
```

(On older computers, the CLEAR key is missing and the CLS command doesn't work. To find out about *your* computer, see the appendix.)



HOW TO PRINT ON PAPER

Now you can make the computer print messages, poems, and business letters on your screen. But if you want to mail those messages and letters to your friends, you have a problem: the post office refuses to mail your screen! And it's hard to stuff your screen into an envelope! So to mail the computer's wise words to your friends, you must tell the computer to print on paper instead of on the screen.

To make the computer print on paper, attach the computer to a *printer*, which looks like a typewriter and handles paper.

You can control the printer in four ways.

The PRINT key. The simplest way is to press the PRINT key, which is on the Walter keyboard. Pressing the PRINT key makes the printer dump onto paper a snapshot of everything that's on the screen. The snapshot on the paper is called a *screen dump*.

The ECHO key. A fancier way to print on paper is to press the ECHO key, which is also on the Walter keyboard. Pressing the ECHO key turns your entire computer system into a gigantic typewriter. After you press the ECHO key, whatever you or the computer type onto the screen will also be echoed onto the paper. For example, if you press the ECHO key and then type the word LOVE, the word LOVE will appear onto the screen and also on the paper. The printer will continue to echo everything that's typed, until you tell the printer to stop echoing. To tell the printer to stop, just press the ECHO key again.

The LPRINT command. If you type LPRINT instead of PRINT, the computer will print on paper instead of on the screen. For example, to make the computer print I LOVE YOU on paper, type this:

```
LPRINT "I LOVE YOU"
```

The LLIST command. To list your program on paper, instead of on the screen, type an extra L in front of the word LIST, like this:

```
LLIST
```

Differences. Although modern computers include a PRINT key and an ECHO key and understand LPRINT and LLIST, older computers do not. To find out about *your* computer, read the appendix.

GOING & STOPPING

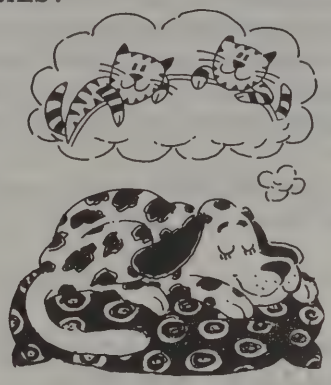
GO TO

This program makes the computer go wild:

```
10 PRINT "CAT"
20 PRINT "DOG"
30 GO TO 10
```

Line 10 makes the computer print CAT. Line 20 makes it print DOG. Line 30 makes it go back to line 10 again, so it prints CAT again. Then it prints DOG again, and comes to line 30 again, which makes it go back to line 10 again, print CAT and DOG again, and then jump back again. . . . The computer will print the words CAT and DOG again and again, like this:

```
CAT
DOG
CAT
DOG
CAT
DOG
CAT
DOG
CAT
DOG
etc.
```



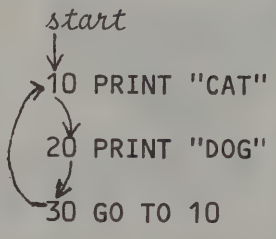
The computer will try to print the words CAT and DOG again and again, *forever*. Try running that program; you'll have fun watching the computer go crazy. The computer will print the words CAT and DOG on every line of your screen; and even when all the lines of your screen have been used, the computer still doesn't stop: all the CATs and DOGs rise to the top of the screen and finally off the screen, to make way for new CATs and DOGs to appear at the bottom of the screen. You'll see a blur, as thousands of CATs and DOGs go flying up the screen.

If you told the computer to use the printer (which prints on paper), you'll see reams of paper spewing out, saying CAT and DOG; soon the whole floor will be buried under piles of paper that say CAT and DOG thousands of times!

To stop this madness, you must give the computer some kind of "jolt" that will put it out of its misery. And now we come to the controversial part of this book. To put the computer out of its misery, you must give it an *abortion*. To give the abortion, press the BREAK key. That makes the computer stop running your program; it will *abort your program* and say OK. Then you can give any command, such as LIST.

(The BREAK key is on the Walter keyboard. To find out about *your* computer's keyboard, check the appendix.)

Here's a picture of that program:



The computer follows the arrows, which make it go round and round in a *loop*. Since the computer will try to go round and round the loop forever, the loop is called *infinite*. The only way to stop an infinite loop is to abort it.

In that program, you typed just three lines; but since the third line said to GO TO the beginning, the computer does an infinite loop. By saying GO TO, you can make the computer do an infinite amount of work. Moral: *the computer can turn a finite amount of human energy into an infinite amount of good.* Putting it another way: *the computer can multiply your abilities by infinity.*

Love. Let's spread love throughout the world! Type this program; it makes the computer multiply your love:

```
10 PRINT "LOVE"
20 GO TO 10
```

Even though you typed the word LOVE just once, the computer will print it many times, like this:

```
LOVE
LOVE
LOVE
LOVE
LOVE
etc.
```

The computer will print repeatedly, until you abort its love.

For more lovely fun, put a semicolon after LOVE, like this:

```
10 PRINT "LOVE";
20 GO TO 10
```

The semicolon makes the computer print LOVE *next to* LOVE:

```
LOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVE
LOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVE
LOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVELOVE
etc.
```

If you put a space after LOVE, like this—

```
10 PRINT "LOVE ";
20 GO TO 10
```

the computer will print a space after each LOVE:

```
LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE
LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE
LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE LOVE
etc.
```

For an even wilder love orgy, put *nine* spaces after LOVE:

```
10 PRINT "LOVE         ";
20 GO TO 10
```

The computer will print:

```
LOVE         LOVE         LOVE         LOVE         LOVE         LOVE         LO
VE         LOVE         LOVE         LOVE         LOVE         LOVE         LOVE
         LOVE         LOVE         LOVE         LOVE         LOVE         LOVE
       LOVE         LOVE         LOVE         LOVE         LOVE         LOVE
     LOVE         LOVE         LOVE         LOVE         LOVE         LOVE
etc.
```

Instead of putting *nine* spaces after LOVE, try a different number of spaces, and see what happens. Experiment! You'll get strange optical illusions, where the word LOVE seems to march across the screen, or blink, or even hop!

Instead of making the computer print LOVE, try making it print an advertisement (such as BUY JOE'S MEAT), or a political slogan (such as STOP POLLUTION), or your name, or something else that makes you feel emotional. Like a dog, the computer imitates its master's personality. If your computer acts "cold and heartless", it's because *you* are.

Smelly poetry. Suppose you want to send this poem to all your friends:

```
I'M HAVING TROUBLE
WITH MY NOSE.
THE ONLY THING IT DOES IS:
BLOWS!
```

Just run this program:

```
10 PRINT "I'M HAVING TROUBLE"
20 PRINT "WITH MY NOSE."
30 PRINT "THE ONLY THING IT DOES IS:"
40 PRINT "BLOWS!"
50 PRINT
60 GO TO 10
```

Lines 10-40 print the poem. Line 50 prints a blank line at the end of the poem. Line 60 makes the computer do all that repeatedly, so the computer will print:

```
I'M HAVING TROUBLE
WITH MY NOSE.
THE ONLY THING IT DOES IS:
BLOWS!

I'M HAVING TROUBLE
WITH MY NOSE.
THE ONLY THING IT DOES IS:
BLOWS!

I'M HAVING TROUBLE
WITH MY NOSE.
THE ONLY THING IT DOES IS:
BLOWS!
```

etc.

The computer will print infinitely many copies of the poem—unless you abort it. If you activate the computer's printer, the copies will be printed on paper, so you can mail them to your nosy friends.

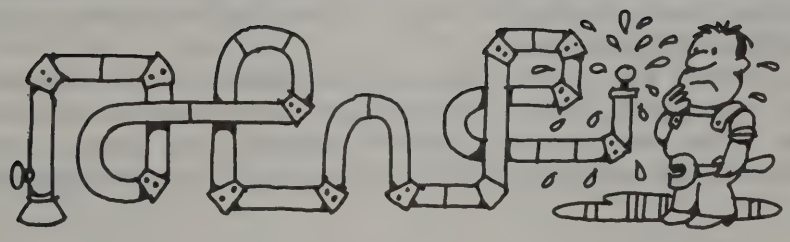
LIFE CAN BE AN INFINITE LOOP

A program that makes the computer do the same thing again and again forever is an infinite loop. Some humans act just like computers. Those humans do the same thing again and again. Every morning they GO TO work, and every evening they GO TO home. GO TO work, GO TO home, GO TO work, GO TO home, GO TO work, GO TO home, . . . Their lives are sheer drudgery. They're caught in an infinite loop.

Go to your bathroom, get your bottle of shampoo, and look at the instructions on the back. A typical bottle has three instructions:

Lather.
Rinse.
Repeat.

Those instructions say to lather, then rinse, then repeat—which means to lather again, then rinse again, then repeat again—which means to lather again, then rinse again, then repeat again—which means to lather again, then rinse again, then repeat again. . . . If you follow those instructions, you'll never finish washing your hair! The instructions are an infinite loop! The instructions are a program: they program you to use lots of shampoo! That's how infinite loops help sell shampoo.



INFINITE PAUSE

This program tortures the computer:

```
10 GO TO 10
```

Line 10 tells the computer to go to line 10, which tells the computer to go to line 10, which tells the computer to go to line 10. . . . The computer will do line 10 again and again, forever. So the computer will spend the rest of its life just mumbling to itself. That program has turned your beautiful computer into a bumbling, mumbling idiot. The "10 GO TO 10" is called an *infinite pause*, because it makes the computer pause from doing all useful work, forever.

If you're mean enough to give your computer that program, please abort it! Otherwise, a member of the Association for Prevention of Cruelty to Computers will stick pins in your voodoo doll.

To make the program even meaner, insert line 5, like this:

```
5 CLS
10 GO TO 10
```

Line 5 makes the computer CLear the Screen, so that the entire screen becomes blank. And line 10 renders the computer completely useless: the computer will ignore anything you type, since the computer is busy doing "10 GO TO 10" forever. You now have a computer whose screen is permanently blank and that ignores anything you type on the keyboard! To be really mean, walk up to your best friend's computer, type that program, type the word RUN, and watch your friend suffer when the screen and the keyboard both stop working! (And see how long your friend remains your "friend"!) The only way your (ex-)friend can fix the computer is to give an abortion or turn off the power.

(Remember: *most* computers understand CLS, but *yours* might not. To find out about *your* computer, check the appendix.)

To elaborate on that program, add line 7, like this:

```
5 CLS
7 PRINT "I REFUSE TO WORK FOR YOU"
10 GO TO 10
```

When you RUN that program, the screen goes blank, then the computer says it refuses to work for you, and then line 10 does indeed make the computer refuse to work for you! The computer will refuse all your work—it'll just mumble to itself—until you give it an abortion.

Now that I've turned you into a computer saboteur, the whole world will worry that you'll kidnap computers, hold them hostage, and threaten to turn them into "10 GO TO 10" idiots! You can cause a moral dilemma for people who are against abortions and are even more against "pulling the plug". Is God against abortions for computers?

HOW TO SKIP AHEAD

Did you ever dream about having a picnic in the woods? This program expresses that dream:

```
10 PRINT "LET'S MUNCH"
20 PRINT "SANDWICHES UNDER"
30 PRINT "THE TREES"
```

It makes the computer print:

```
LET'S MUNCH
SANDWICHES UNDER
THE TREES
```

Let's turn that dream into a nightmare, in which we all become giant termites. To do that, insert line 15, like this:

```
          10 PRINT "LET'S MUNCH"
New line→→15 GO TO 30
          20 PRINT "SANDWICHES UNDER"
          30 PRINT "THE TREES"
```

The computer does line 10, which prints LET'S MUNCH. Then it does line 15, which makes it skip ahead to line 30, which prints THE TREES. So altogether, the computer prints:

```
LET'S MUNCH
THE TREES
```

In that program, line 15 made the computer skip ahead to line 30. It made the computer skip over line 20.

END

To make the computer skip the bottom part of your program, say END:

```
10 PRINT "SHE SMELLS"
20 END
30 PRINT "OF PERFUME"
```

The computer will print SHE SMELLS and then end, without printing OF PERFUME.

After the computer ends, it'll say OK. Then you can type any command, such as LIST (to see the program again) or RUN (to make the computer print SHE SMELLS again) or NEW (to create a new program).

STOP

Instead of saying END, you can say STOP. For example, you can say:

```
10 PRINT "SHE SMELLS"
20 STOP
30 PRINT "OF PERFUME"
```

The computer will print SHE SMELLS and then stop, without printing OF PERFUME.

When the computer encounters the word STOP, it prints a message telling you which line said STOP. For example, when the computer stops because line 20 said STOP, the computer prints "STOPPED AT LINE 20" or "BREAK IN LINE 20" or "BREAK IN 20" (the exact wording depends on which computer you bought).

So if you type—

```
10 PRINT "SHE SMELLS"
20 STOP
30 PRINT "OF PERFUME"
```

and then type RUN, the computer will print something like this:

```
SHE SMELLS
BREAK IN 20
```

VARIABLES

NUMERIC VARIABLES

A letter can stand for a number. For example, X can stand for the number 47, as in this program:

```
10 X=47
20 PRINT X+2
```

Line 10 says X stands for the number 47. In other words, X is a name for the number 47. Line 20 says to print X+2; since X is 47, X+2 is 49; so the computer will print 49. That's the only number the computer will print; it will not print 47.

A letter that stands for a number is called a *numeric variable*. In that program, X is a numeric variable; it stands for the number 47. The *value* of X is 47. Line 10 is called an *assignment statement*, because it *assigns* 47 to X.

Another example:

```
10 Y=38
20 PRINT Y-2
```

Line 10 says Y is a numeric variable that stands for the number 38. Line 20 says to print Y-2; since Y is 38, Y-2 is 36; so the computer will print 36.

Example:

```
10 B=8
20 PRINT B*3
```

Line 10 says B is 8. Line 20 says to print B*3, which is 8*3, which is 24; so the computer will print 24.

One variable can define another:

```
10 M=6
20 P=M+1
30 PRINT M*P
```

Line 10 says M is 6. Line 20 says P is M+1, which is 6+1, which is 7; so P is 7. Line 30 says to print M*P, which is 6*7, which is 42; so the computer will print 42.

A value can change:

```
10 F=4
20 F=9
30 PRINT F*2
```

Line 10 says F's value is 4. Line 20 changes F's value to 9, so line 30 prints 18.

On the left side of the equal sign, you must have *one* variable:

Allowed	Not allowed	Not allowed
P=M+1	P-M=1	1=P-M
↑	↑ ↑	↑
one variable	*two variables*	*not a variable*

The variable on the left side of the equation is the only one that changes. For example, the statement P=M+1 changes the value of P but not M. The statement A=B changes the value of A but not B:

```
10 A=1
20 B=7
30 A=B
40 PRINT A+B
```

Line 30 changes A, to make it equal B; so A becomes 7. Since both A and B are now 7, line 40 prints 14.

"A=B" has a different effect than "B=A". That's because "A=B" changes the value of A (but not B); "B=A" changes the value of B (but not A). Compare these programs:

```
10 A=1              10 A=1
20 B=7              20 B=7
30 A=B              30 B=A
40 PRINT A+B        40 PRINT A+B
```

In the left program (which you saw before), line 30 changes A to 7, so both A and B are 7; line 40 prints 14. In the right program, line 30 changes B to 1, so both A and B are 1; line 40 prints 2.

If you don't define a numeric variable, the computer assumes it's zero:

```
10 PRINT R
```

Since R hasn't been defined, line 10 prints zero.

The computer doesn't look ahead:

```
10 PRINT J
20 J=5
```

When the computer encounters line 10, it doesn't look ahead to find out what J is. As of line 10, J is still undefined, so the computer prints zero.

WHEN TO USE VARIABLES

When should you use variables? Here's a practical example. . . .

Suppose you're selling something that costs $1297.43, and you want to do these calculations:

multiply $1297.43 by 2
multiply $1297.43 by .05
add $1297.43 to $483.19
divide $1297.43 by 37
subtract $1297.43 from $8598.61
multiply $1297.43 by 28.7

To do those six calculations, you could run this program:

```
10 PRINT 1297.43*2; 1297.43*.05; 1297.43+483.19; 1297.43/37; 8598.61-1297.43
20 PRINT 1297.43*28.7
```

But that program's silly, since it contains the number 1297.43 six times. This program's briefer, because it uses a variable:

```
10 C=1297.43
20 PRINT C*2; C*.05; C+483.19; C/37; 8598.61-C; C*28.7
```

So whenever you need to use a number several times, turn the number into a variable; it will make your program briefer.

STRING VARIABLES

A string is any collection of characters, such as "I LOVE YOU". Each string must be in quotation marks. A letter can stand for a string—if you put a dollar sign after the letter, like this:

```
10 G$="DOWN"
20 PRINT G$
```

Line 10 says G$ stands for the string "DOWN". Line 20 prints:

```
DOWN
```

In that program, G$ is a variable. Since it stands for a string, it's called a *string variable*. Every string variable must end with a dollar sign; the dollar sign is supposed to remind you of a fancy S, which stands for String. Line 10 is pronounced, "G String is DOWN".

If you're paranoid, you'll love this program:

```
10 L$="THEY'RE LAUGHING AT YOU"
20 PRINT L$
30 PRINT L$
40 PRINT L$
```

Line 10 says L$ stands for the string "THEY'RE LAUGHING AT YOU". Lines 20, 30, and 40 make the computer print:

```
THEY'RE LAUGHING AT YOU
THEY'RE LAUGHING AT YOU
THEY'RE LAUGHING AT YOU
```

Nursery rhymes. The computer can recite nursery rhymes:

```
10 P$="PEAS PORRIDGE"
20 PRINT P$;" HOT"
30 PRINT P$;" COLD"
40 PRINT P$;" IN THE POT"
50 PRINT "NINE DAYS OLD"
```

Line 10 says P$ stands for "PEAS PORRIDGE". Lines 20-50 make the computer print:

```
PEAS PORRIDGE HOT
PEAS PORRIDGE COLD
PEAS PORRIDGE IN THE POT
NINE DAYS OLD
```

This program prints a fancier rhyme:

```
10 H$="HICKORY, DICKORY, DOCK!"
20 M$="THE MOUSE (SQUEAK! SQUEAK!)"
30 C$="THE CLOCK (TICK! TOCK!)"
40 PRINT H$
50 PRINT M$;" RAN UP ";C$
60 PRINT C$;" STRUCK ONE"
70 PRINT M$;" RAN DOWN"
80 PRINT H$
```

Lines 10-30 define H$, M$, and C$. Lines 40-80 make the computer print:

```
HICKORY, DICKORY, DOCK!
THE MOUSE (SQUEAK! SQUEAK!) RAN UP THE CLOCK (TICK! TOCK!)
THE CLOCK (TICK! TOCK!) STRUCK ONE
THE MOUSE (SQUEAK! SQUEAK!) RAN DOWN
HICKORY, DICKORY, DOCK!
```

Undefined strings. If you don't define a string variable, the computer assumes it's blank:

```
10 PRINT F$
```

Since F$ hasn't been defined, line 10 makes the computer print a line that says nothing; the line the computer prints is blank.

LONG VARIABLE NAMES

A numeric variable's name can be a letter (such as X) or a longer combination of characters (such as PROFIT.IN.1986.BEFORE.NOVEMBER.PROMOTION). For example, you can type:

```
10 PROFIT.IN.1986.BEFORE.NOVEMBER.PROMOTION = 3497.18
20 PROFIT.IN.1986 = PROFIT.IN.1986.BEFORE.NOVEMBER.PROMOTION + 6214.27
30 PRINT PROFIT.IN.1986
```

The computer will print:

```
 9711.45
```

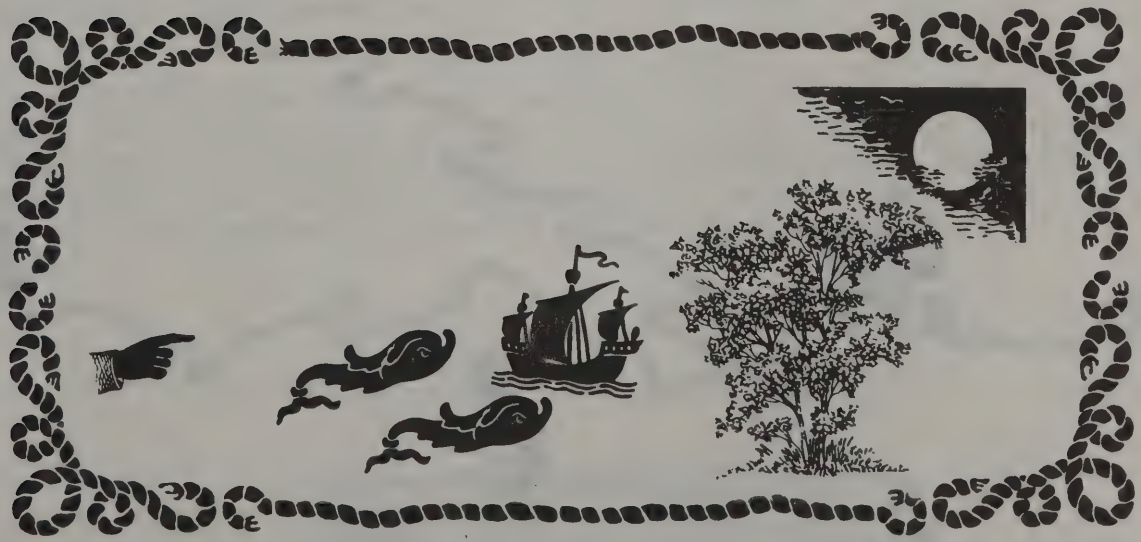
The variable's name can be quite long: up to 40 characters! The first character in the name must be a letter; the remaining characters can be letters, digits, or periods. The name must not be a word that has a special meaning to the computer; for example, the name cannot be PRINT or GO or TO.

If the variable stands for a string, the name can have up to 40 characters, followed by a dollar sign, making a total of 41 characters, like this:
MY.JOB.IN.1986.BEFORE.NOVEMBER.PROMOTION$

Although modern computers permit long variable names, primitive computers don't. To find out whether *your* computer permits long variable names, check the appendix.

Professional programmers use long variable names, because long names make the programs easier to understand.

I'll avoid long names, since some computers don't permit them; but if your computer permits them, use them!



INPUT

STRING INPUT

Humans ask questions. So if you want to turn the computer into a human, you must make it ask questions too. To make the computer ask a question, use the word INPUT. This program makes the computer ask for your name:

```
10 INPUT "WHAT IS YOUR NAME";N$
20 PRINT "I ADORE ANYONE WHOSE NAME IS ";N$
```

When the computer sees line 10, the computer asks "WHAT IS YOUR NAME?" and then waits for you to answer the question; your answer will be called N$. For example, if you answer MARIA, then N$ is MARIA. Line 20 makes the computer print:

```
I ADORE ANYONE WHOSE NAME IS MARIA
```

Here's the whole conversation; I've underlined the parts typed by you:

You tell the computer to run→→→→→RUN
The computer asks for your name→→WHAT IS YOUR NAME? MARIA
The computer praises your name→→→I ADORE ANYONE WHOSE NAME IS MARIA

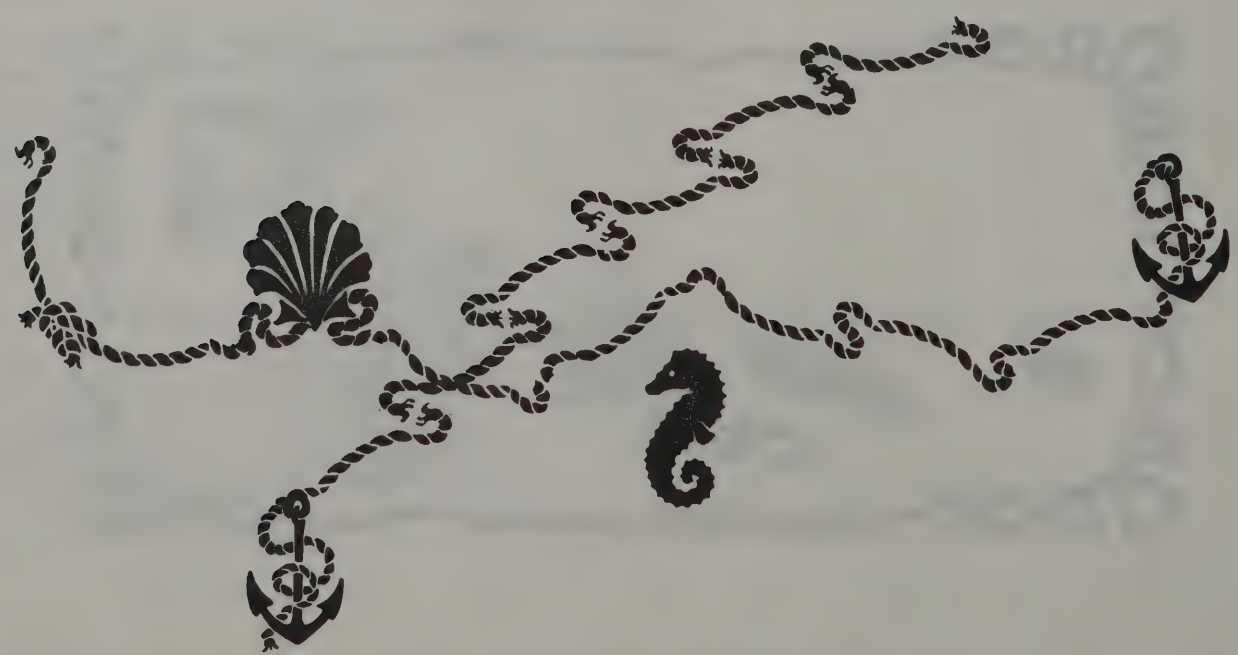
Try that example. Be careful! When you type line 10, which says INPUT, make sure you type the two quotation marks and the semicolon. You don't have to type a question mark; when the computer runs your program, it will automatically put a question mark at the end of the question.

Just for fun, run that program again and pretend you're somebody else:

You tell the computer to run→→→→→RUN
The computer asks for your name→→WHAT IS YOUR NAME? BUD
The computer praises your name→→→I ADORE ANYONE WHOSE NAME IS BUD

When the computer asks for your name, if you say something weird, the computer will give you a weird reply:

Make the computer run→→→→→RUN
Computer asks your name→→→WHAT IS YOUR NAME? NONE OF YOUR BUSINESS!!!
The computer replies→→→→→→I ADORE ANYONE WHOSE NAME IS NONE OF YOUR BUSINESS!!!



COLLEGE ADMISSIONS

This program prints a letter, admitting you to the college of your choice:

```
10 INPUT "WHAT COLLEGE WOULD YOU LIKE TO ENTER";C$
20 PRINT "CONGRATULATIONS!"
30 PRINT "YOU HAVE JUST BEEN ADMITTED TO ";C$
40 PRINT "IT FITS YOUR PERSONALITY"
50 PRINT "I HOPE YOU GO TO ";C$
60 PRINT "                RESPECTFULLY YOURS"
70 PRINT "                THE DEAN OF ADMISSIONS"
```

Line 10 makes the computer ask "WHAT COLLEGE WOULD YOU LIKE TO ENTER?" and then wait for you to answer; your answer will be called C$. If you'd like to be admitted to HARVARD, you'll be pleased:

```
You tell the computer to run→→RUN
The computer asks→→→→→→→→→→→→→WHAT COLLEGE WOULD YOU LIKE TO ENTER? HARVARD
The computer admits you→→→→→→→CONGRATULATIONS!
                              YOU HAVE JUST BEEN ADMITTED TO HARVARD
                              IT FITS YOUR PERSONALITY
                              I HOPE YOU GO TO HARVARD
                                        RESPECTFULLY YOURS
                                        THE DEAN OF ADMISSIONS
```

You can choose any college you wish:

```
RUN
WHAT COLLEGE WOULD YOU LIKE TO ENTER? HELL
CONGRATULATIONS!
YOU HAVE JUST BEEN ADMITTED TO HELL
IT FITS YOUR PERSONALITY
I HOPE YOU GO TO HELL
          RESPECTFULLY YOURS
          THE DEAN OF ADMISSIONS
```

That program consists of three parts:

1. The computer begins by asking you a question ("What college would you like to be admitted to?"). The computer's question is called the *prompt*, because it prompts you to answer.

2. Your answer (the college's name) is called *your input*, because it's information that you're *putting into* the computer.

3. The computer's reply (the admission letter) is called the *computer's output*, because it's the final answer that the computer puts out.

INPUT VERSUS PRINT

The word INPUT is the opposite of the word PRINT. The word PRINT makes the computer print information out; the word INPUT makes the computer take information in. The word PRINT makes the computer output; the word INPUT makes the computer accept your input. Input and Output are collectively called *I/O*, so the INPUT and PRINT statements are called *I/O statements*.

STORIES

Let's make the computer write a story, by filling in the blanks:

```
ONCE UPON A TIME, THERE WAS A YOUNGSTER NAMED _____
                                              your name
WHO HAD A FRIEND NAMED _____
                       friend's name
_____ WANTED TO _____ _____
  your name          verb (such as "pat")   friend's name

BUT _____ DIDN'T WANT TO _____ _____
     friend's name          verb (such as "pat")   your name

WILL _____ _____ _____
      your name   verb (such as "pat")   friend's name

WILL _____ _____ _____
      friend's name   verb (such as "pat")   your name

TO FIND OUT, COME BACK AND SEE THE NEXT EXCITING EPISODE

OF _____ AND _____
     your name      friend's name
```

To write the story, the computer must ask for your name, your friend's name, and a verb. To make the computer ask, your program must say INPUT:

```
10 INPUT "WHAT'S YOUR NAME";Y$
20 INPUT "WHAT'S YOUR FRIEND'S NAME";F$
30 INPUT "IN 1 WORD, SAY SOMETHING YOU CAN DO TO YOUR FRIEND";V$
```

Then make the computer print the story:

```
40 PRINT "HERE'S MY STORY...."
50 PRINT "ONCE UPON A TIME, THERE WAS A YOUNGSTER NAMED ";Y$
60 PRINT "WHO HAD A FRIEND NAMED ";F$
70 PRINT Y$;" WANTED TO ";V$;" ";F$
80 PRINT "BUT ";F$;" DIDN'T WANT TO ";V$;" ";Y$
90 PRINT "WILL ";Y$;" ";V$;" ";F$
100 PRINT "WILL ";F$;" ";V$;" ";Y$
110 PRINT "TO FIND OUT, COME BACK AND SEE THE NEXT EXCITING EPISODE"
120 PRINT "OF ";Y$;" AND ";F$
```

Here's a sample run:

```
WHAT'S YOUR NAME? DRACULA
WHAT'S YOUR FRIEND'S NAME? MARILYN MONROE
IN 1 WORD, SAY SOMETHING YOU CAN DO TO YOUR FRIEND? BITE
HERE'S MY STORY....
ONCE UPON A TIME, THERE WAS A YOUNGSTER NAMED DRACULA
WHO HAD A FRIEND NAMED MARILYN MONROE
DRACULA WANTED TO BITE MARILYN MONROE
BUT MARILYN MONROE DIDN'T WANT TO BITE DRACULA
WILL DRACULA BITE MARILYN MONROE
WILL MARILYN MONROE BITE DRACULA
TO FIND OUT, COME BACK AND SEE THE NEXT EXCITING EPISODE
OF DRACULA AND MARILYN MONROE
```

Here's another run:

```
WHAT'S YOUR NAME? SUPERMAN
WHAT'S YOUR FRIEND'S NAME? KING KONG
IN 1 WORD, SAY SOMETHING YOU CAN DO TO YOUR FRIEND? TICKLE
HERE'S MY STORY....
ONCE UPON A TIME, THERE WAS A YOUNGSTER NAMED SUPERMAN
WHO HAD A FRIEND NAMED KING KONG
SUPERMAN WANTED TO TICKLE KING KONG
BUT KING KONG DIDN'T WANT TO TICKLE SUPERMAN
WILL SUPERMAN TICKLE KING KONG
WILL KING KONG TICKLE SUPERMAN
TO FIND OUT, COME BACK AND SEE THE NEXT EXCITING EPISODE
OF SUPERMAN AND KING KONG
```

Try it: put in your own name, the name of your friend, and something you'd like to do to your friend.

This program creates a story that's fancier:

```
10 INPUT "WHAT'S YOUR NAME";Y$
20 INPUT "WHAT'S YOUR FRIEND'S NAME";F$
30 INPUT "WHAT'S THE NAME OF ANOTHER FRIEND";A$
40 INPUT "NAME A COLOR";C$
50 INPUT "NAME A PLACE";P$
60 INPUT "NAME A FOOD";D$
70 INPUT "NAME AN OBJECT";J$
80 INPUT "NAME A PART OF THE BODY";B$
90 INPUT "NAME A STYLE OF COOKING (SUCH AS BAKED OR FRIED)";S$
100 PRINT
110 PRINT "CONGRATULATIONS ";Y$
120 PRINT "YOU'VE WON THE BEAUTY CONTEST, BECAUSE OF YOUR GORGEOUS ";B$
130 PRINT "YOUR PRIZE IS A ";C$;" ";J$
140 PRINT "PLUS A TRIP TO ";P$;" WITH YOUR FRIEND ";F$
150 PRINT "PLUS...AND THIS IS THE BEST PART OF ALL..."
160 PRINT "DINNER FOR THE TWO OF YOU AT ";A$;"'S NEW RESTAURANT"
170 PRINT "WHERE ";A$;" WILL GIVE YOU ALL THE ";S$;" ";D$;" YOU CAN EAT"
180 PRINT "CONGRATULATIONS ";Y$;"...TODAY'S YOUR LUCKY DAY..."
190 PRINT "NOW EVERYBODY WANTS TO KISS YOUR AWARD-WINNING ";B$
```

Here's a sample run:

```
WHAT'S YOUR NAME? LONG JOHN SILVER
WHAT'S YOUR FRIEND'S NAME? THE PARROT
WHAT'S THE NAME OF ANOTHER FRIEND? JIM
NAME A COLOR? GOLD
NAME A PLACE? TREASURE ISLAND
NAME A FOOD? RUM-SOAKED COCONUTS
NAME AN OBJECT? CHEST OF JEWELS
NAME A PART OF THE BODY? MISSING LEG
NAME A STYLE OF COOKING (SUCH AS BAKED OR FRIED)? BARBECUED

CONGRATULATIONS LONG JOHN SILVER
YOU'VE WON THE BEAUTY CONTEST, BECAUSE OF YOUR GORGEOUS MISSING LEG
YOUR PRIZE IS A GOLD CHEST OF JEWELS
PLUS A TRIP TO TREASURE ISLAND WITH YOUR FRIEND THE PARROT
PLUS...AND THIS IS THE BEST PART OF ALL...
DINNER FOR THE TWO OF YOU AT JIM'S NEW RESTAURANT
WHERE JIM WILL GIVE YOU ALL THE BARBECUED RUM-SOAKED COCONUTS YOU CAN EAT
CONGRATULATIONS LONG JOHN SILVER...TODAY'S YOUR LUCKY DAY...
NOW EVERYBODY WANTS TO KISS YOUR AWARD-WINNING MISSING LEG
```

This run describes the contest that brought Ronald Reagan to the White House:

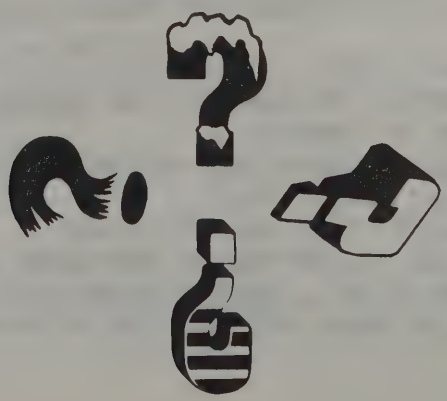
```
WHAT'S YOUR NAME? RONNIE REAGAN
WHAT'S YOUR FRIEND'S NAME? NANCY
WHAT'S THE NAME OF ANOTHER FRIEND? ALICE
NAME A COLOR? RED-WHITE-AND-BLUE
NAME A PLACE? THE WHITE HOUSE
NAME A FOOD? JELLY BEANS
NAME AN OBJECT? COWBOY HAT
NAME A PART OF THE BODY? CHEEKS
NAME A STYLE OF COOKING (SUCH AS BAKED OR FRIED)? STEAMED

CONGRATULATIONS RONNIE REAGAN
YOU'VE WON THE BEAUTY CONTEST, BECAUSE OF YOUR GORGEOUS CHEEKS
YOUR PRIZE IS A RED-WHITE-AND-BLUE COWBOY HAT
PLUS A TRIP TO THE WHITE HOUSE WITH YOUR FRIEND NANCY
PLUS...AND THIS IS THE BEST PART OF ALL...
DINNER FOR THE TWO OF YOU AT ALICE'S NEW RESTAURANT
WHERE ALICE WILL GIVE YOU ALL THE STEAMED JELLY BEANS YOU CAN EAT
CONGRATULATIONS RONNIE REAGAN...TODAY'S YOUR LUCKY DAY...
NOW EVERYBODY WANTS TO KISS YOUR AWARD-WINNING CHEEKS
```

BILLS

If you're a nasty bill collector, you'll love this program:

```
10 INPUT "WHAT IS THE CUSTOMER'S FIRST NAME";F$
20 INPUT "LAST NAME";L$
30 INPUT "STREET ADDRESS";A$
40 INPUT "CITY";C$
50 INPUT "STATE";S$
60 INPUT "ZIP CODE";Z$
70 PRINT
80 PRINT F$;" ";L$
90 PRINT A$
100 PRINT C$;" ";S$;" ";Z$
110 PRINT
120 PRINT "DEAR ";F$
130 PRINT "   YOU STILL HAVEN'T PAID THE BILL."
140 PRINT "IF YOU DON'T PAY IT SOON, ";F$;","
150 PRINT "I'LL COME VISIT YOU IN ";C$
160 PRINT "AND PERSONALLY SHOOT YOU."
170 PRINT "            YOURS TRULY,"
180 PRINT "            SURE-AS-SHOOTIN'"
190 PRINT "            YOUR CRAZY CREDITOR"
```

Can you figure out what that program does?



NUMERIC INPUT

This program makes the computer predict your future:

```
10 PRINT "I WILL PREDICT WHAT WILL HAPPEN TO YOU IN THE YEAR 2000!"
20 INPUT "IN WHAT YEAR WERE YOU BORN";Y
30 PRINT "IN THE YEAR 2000, YOU WILL BECOME";2000-Y;"YEARS OLD."
```

Here's a sample run:

```
I WILL PREDICT WHAT WILL HAPPEN TO YOU IN THE YEAR 2000!
IN WHAT YEAR WERE YOU BORN? 1962
IN THE YEAR 2000, YOU WILL BECOME 38 YEARS OLD.
```

Suppose you're selling tickets to a play. Each ticket costs $2.79. (You decided $2.79 would be a nifty price, because the cast has 279 people.) This program finds the price of multiple tickets:

```
10 INPUT "HOW MANY TICKETS";T
20 PRINT "THE TOTAL PRICE IS $";T*2.79
```

This program tells you how much the "energy crisis" costs you, when you drive your car:

```
10 INPUT "HOW MANY MILES DO YOU WANT TO DRIVE";M
20 INPUT "HOW MANY PENNIES DOES A GALLON OF GAS COST";P
30 INPUT "HOW MANY MILES-PER-GALLON DOES YOUR CAR GET";R
40 PRINT "THE GAS FOR YOUR TRIP WILL COST YOU $";M*P/(R*100)
```

Here's a sample run:

```
HOW MANY MILES DO YOU WANT TO DRIVE? 300
HOW MANY PENNIES DOES A GALLON OF GAS COST? 127.9
HOW MANY MILES-PER-GALLON DOES YOUR CAR GET? 31
THE GAS FOR YOUR TRIP WILL COST YOU $ 12.3774
```

CONVERSION

This program converts feet to inches:

```
10 INPUT "HOW MANY FEET";F
20 PRINT F;"FEET =";F*12;"INCHES"
```

Here's a sample run:

```
HOW MANY FEET? 3
 3 FEET = 36 INCHES
```

Try to convert to the metric system? This program converts inches to centimeters:

```
10 INPUT "HOW MANY INCHES";I
20 PRINT I;"INCHES =";I*2.54;"CENTIMETERS"
```

Nice day today, isn't it? This program converts the temperature from Celsius to Fahrenheit:

```
10 INPUT "HOW MANY DEGREES CELSIUS";C
20 PRINT C;"DEGREES CELSIUS =";C*1.8+32;"DEGREES FAHRENHEIT"
```

Here's a sample run:

```
HOW MANY DEGREES CELSIUS? 20
 20 DEGREES CELSIUS = 68 DEGREES FAHRENHEIT
```

See, you can write the *Guide* yourself! Just hunt through any old math or science book, find any old formula (such as F=C*1.8+32), and turn it into a program.

IF... THEN

HOW TO TURN YOUR COMPUTER INTO A THERAPIST

Let's turn the computer into a therapist. To begin, make the computer ask the patient, "HOW ARE YOU?" To make the computer ask that question, begin the program like this:

```
10 INPUT "HOW ARE YOU";A$
```

That line makes the computer ask HOW ARE YOU and wait for the patient's answer, which is called A$.

If the patient feels FINE, let's make the computer say THAT'S GOOD. Here's how:

```
20 IF A$="FINE" THEN PRINT "THAT'S GOOD"
```

That line says that if the patient's answer (A$) is FINE, the computer will print THAT'S GOOD.

If the patient feels LOUSY instead, let's make the computer say TOO BAD. Here's how:

```
30 IF A$="LOUSY" THEN PRINT "TOO BAD"
```

Here's the entire program:

```
10 INPUT "HOW ARE YOU";A$
20 IF A$="FINE" THEN PRINT "THAT'S GOOD"
30 IF A$="LOUSY" THEN PRINT "TOO BAD"
```

When the patient types RUN, line 10 makes the computer ask HOW ARE YOU and wait for the patient's answer, which is called A$. If the patient's answer is FINE, line 20 makes the computer reply THAT'S GOOD. If the patient's answer is LOUSY, line 30 makes the computer reply TOO BAD.

Try running that program! Here's a sample run:

```
RUN
HOW ARE YOU? FINE
THAT'S GOOD
```

Here's another run:

```
RUN
HOW ARE YOU? LOUSY
TOO BAD
```

Whenever you say IF, you should also say THEN. For example, line 20 (which says IF) also says THEN. Similarly, line 30 (which says IF) says THEN.

What comes between IF and THEN is called the *condition*. In line 20, the condition is:

```
A$="FINE"
```

If that condition is true (if A$ really does equal "FINE"), the computer does what comes after the word THEN, which is called the *consequence*, which is:

```
PRINT "THAT'S GOOD"
```

In that program, what happens if the patient says some word other that FINE or LOUSY?

For example, what happens if the patient says TERRIBLE? The program doesn't tell the computer how to react to TERRIBLE, so the computer won't print any reaction at all. The computer will just do what it always does at the end of a program: it will say OK (or READY or a similar word).

Let's add extra lines at the bottom of the program, so that if the patient says TERRIBLE, the computer will say TOUGH TURKEY, and so that if the patient says GREAT instead, the computer will say OH WOW:

```
        10 INPUT "HOW ARE YOU";A$
        20 IF A$="FINE" THEN PRINT "THAT'S GOOD"
        30 IF A$="LOUSY" THEN PRINT "TOO BAD"
New line→→40 IF A$="TERRIBLE" THEN PRINT "TOUGH TURKEY"
New line→→50 IF A$="GREAT" THEN PRINT "OH WOW"
```

But what happens if the patient says neither FINE nor LOUSY nor TERRIBLE nor GREAT? Let's make the computer say I FEEL THE SAME WAY:

```
        10 INPUT "HOW ARE YOU";A$
        20 IF A$="FINE" THEN PRINT "THAT'S GOOD"
        30 IF A$="LOUSY" THEN PRINT "TOO BAD"
        40 IF A$="TERRIBLE" THEN PRINT "TOUGH TURKEY"
        50 IF A$="GREAT" THEN PRINT "OH WOW"
New line→→60 PRINT "I FEEL THE SAME WAY"
```

Since line 60 does *not* contain the word IF, the computer will *always* print I FEEL THE SAME WAY at the end of the conversation, regardless of what the patient says.

Here's a sample run:

```
RUN
HOW ARE YOU? FINE
THAT'S GOOD
I FEEL THE SAME WAY
```

Here's another:

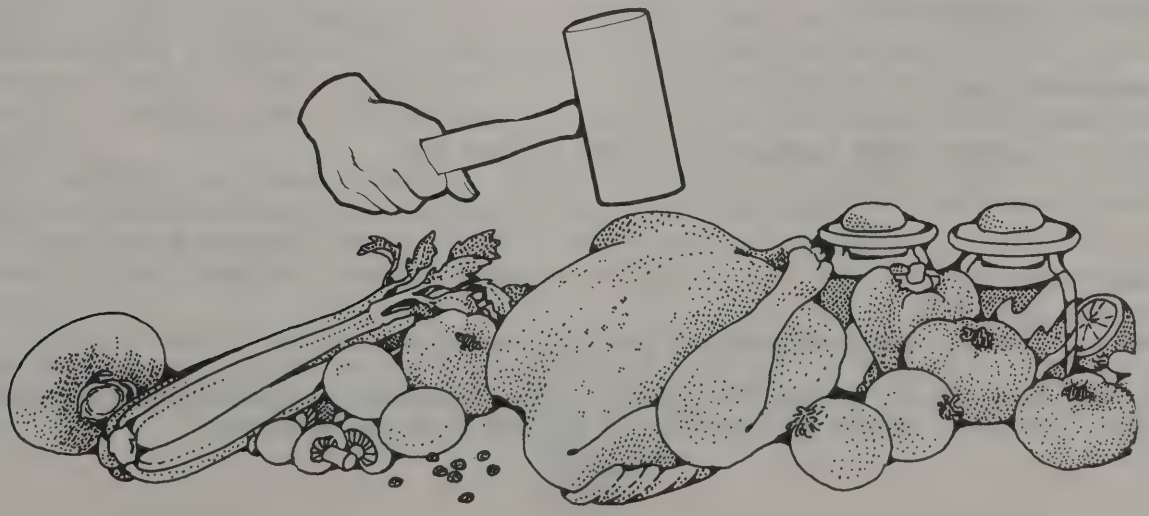
```
RUN
HOW ARE YOU? TERRIBLE
TOUGH TURKEY
I FEEL THE SAME WAY
```

Another:

```
RUN
HOW ARE YOU? LONELY
I FEEL THE SAME WAY
```

Another:

```
RUN
HOW ARE YOU? I DON'T WANT TO TALK WITH YOU
I FEEL THE SAME WAY
```



Having the computer always say I FEEL THE SAME WAY is boring. To make the program more interesting, let's make the computer say I FEEL THE SAME WAY only if the patient says neither FINE nor LOUSY nor TERRIBLE nor GREAT.

So if the patient says FINE, LOUSY, TERRIBLE, or GREAT, let's prevent the computer from saying I FEEL THE SAME WAY. Whenever you want to make the computer skip saying I FEEL THE SAME WAY, type the word END, like this:

```
            10 INPUT "HOW ARE YOU";A$
Revised line→20 IF A$="FINE" THEN PRINT "THAT'S GOOD": END
Revised line→30 IF A$="LOUSY" THEN PRINT "TOO BAD": END
Revised line→40 IF A$="TERRIBLE" THEN PRINT "TOUGH TURKEY": END
Revised line→50 IF A$="GREAT" THEN PRINT "OH WOW": END
            60 PRINT "I FEEL THE SAME WAY"
```

In that program, if the patient types FINE or LOUSY or TERRIBLE or GREAT, lines 20-40 make the computer print a two-word message and then END, without printing I FEEL THE SAME WAY. The only way the computer can reach line 60 (which prints I FEEL THE SAME WAY) is if the patient types neither FINE nor LOUSY nor TERRIBLE nor GREAT.

Here's a sample run:

```
RUN
HOW ARE YOU? FINE
THAT'S GOOD
```

Here's another:

```
RUN
HOW ARE YOU? LONELY
I FEEL THE SAME WAY
```

After the computer has given the patient some therapy, let's make the computer charge the patient $50. To do that, add this line:

```
70 PRINT "I HOPE YOU ENJOYED YOUR THERAPY--NOW YOU OWE ME $50"
```

To make sure the computer always goes to that line and collects the $50, regardless of what the patient says, replace each END by GO TO 70. Altogether, the program looks like this:

```
10 INPUT "HOW ARE YOU";A$
20 IF A$="FINE" THEN PRINT "THAT'S GOOD": GO TO 70
30 IF A$="LOUSY" THEN PRINT "TOO BAD": GO TO 70
40 IF A$="TERRIBLE" THEN PRINT "TOUGH TURKEY": GO TO 70
50 IF A$="GREAT" THEN PRINT "OH WOW": GO TO 70
60 PRINT "I FEEL THE SAME WAY"
70 PRINT "I HOPE YOU ENJOYED YOUR THERAPY--NOW YOU OWE ME $50"
```

Here's a sample run:

```
RUN
HOW ARE YOU? FINE
THAT'S GOOD
I HOPE YOU ENJOYED YOUR THERAPY--NOW YOU OWE ME $50
```

Here's another:

```
RUN
HOW ARE YOU? LONELY
I FEEL THE SAME WAY
I HOPE YOU ENJOYED YOUR THERAPY--NOW YOU OWE ME $50
```

In that program, try changing the strings, to make the computer print cleverer remarks, become a better therapist, and charge even more money.

In that program, the only words the computer understands are INPUT, IF, THEN, PRINT, GO, and TO. Those words are called the *keywords*.

By using just those keywords, you can write any program you wish! Let's see why. . . .

Can a computer become President? To become President of the United States, you need four basic skills.

First, you must be a good talker, so you can give effective speeches saying "Vote for me!", express your views, and make folks do what you want.

But even if you're a good talker, you're useless, unless you're also a good listener. You must be able to listen to people's needs, and ask the people, "What can I do for you, so that you'll be happy, and so that you'll vote for me?"

But even if you're a good talker and a good listener, you're still useless, unless you can make decisions. Should you give more money to poor people? Should you bomb the enemy? Which actions should you take, and under what conditions?

But even if you're a good talker and a good listener and a wise decision-maker, you still need one more trait to become President: you must be able to put up with the daily grind of politics. You must, again and again, shake hands, make compromises, and raise funds. You must have the patience to put up with the repetitive monotony of those chores.

So altogether, to become President of the United States, you need to be a good talker, a good listener, a wise decision maker, and a person patient enough to put up with monotonous repetition.

Those are exactly the four qualities that the computer has! The word PRINT turns the computer into a good speech-maker: by using the word PRINT, you can make the computer write whatever speech you wish. The word INPUT turns the computer into a good listener: by using the word INPUT, you can make the computer ask humans lots of questions, to find out who the humans are and what they want. The words IF and THEN turn the computer into a decision maker: the computer can analyze the IF condition, determine whether that condition is true, and act accordingly. And finally, the words GO and TO enable the computer to perform loops, which the computer will repeat patiently.

So by using the words PRINT, INPUT, IF THEN, and GO TO, you can make the computer imitate any intellectual human activity. Those four magic phrases—PRINT, INPUT, IF THEN, and GO TO—are the only phrases you need, to write whatever program you wish!

Yes, you can make the computer imitate the President of the United States, or do your company's payroll, or compose a beautiful poem, or play a perfect game of chess, or contemplate the meaning of life, or act as if it's falling in love, or do whatever other intellectual or emotional task you wish, by using just those four magic phrases. The only question is: how? The Secret Guide to Computers teaches you how, by showing you many examples of programs that do those remarkable things.

What programmers believe. Yes, we programmers believe that all of life can be explained and programmed. We believe all of life can be reduced to just those four phrases: PRINT, INPUT, IF THEN, and GO TO. Programming is the ultimate act of scientific reductionism: programmers reduce all of life scientifically to just four phrases.

In addition to those keywords (PRINT, INPUT, IF, THEN, GO, and TO), the computer understands extra keywords also, such as END.

The extra keywords (such as END) aren't strictly necessary: if they hadn't been invented, you could still write programs without them. But they make programming easier.

A programmer is a person who translates an ordinary English sentence (such as "act like the President of the United States" or "do the payroll") into a series of BASIC statements, using keywords such as PRINT, INPUT, IF THEN, GO TO, and END.

The mysteries of life. Let's dig deeper into the mysteries of PRINT, INPUT, IF THEN, GO TO, and the extra keywords. The deeper we dig, the more you'll wonder: are *you* just a computer, made of flesh instead of wires? Can everything that *you* do be explained in terms of PRINT, INPUT, IF THEN, and GO TO?

By the time you finish The Secret Guide to Computers, you'll know!

MARY POPPINS MEETS FRANKENSTEIN

Let's make the computer interrogate a human. Let's make the computer begin the interrogation by asking whether the human is male or female. If the human answers MALE, let's make the computer say:

```
SO IS FRANKENSTEIN
```

If the human answers FEMALE, let's make the computer say:

```
SO IS MARY POPPINS
```

Here's the program:

```
10 INPUT "ARE YOU MALE OR FEMALE";A$
20 IF A$="MALE" THEN PRINT "SO IS FRANKENSTEIN"
30 IF A$="FEMALE" THEN PRINT "SO IS MARY POPPINS"
```

Here's a sample run:

```
RUN
ARE YOU MALE OR FEMALE? MALE
SO IS FRANKENSTEIN
```

Here's another:

```
RUN
ARE YOU MALE OR FEMALE? FEMALE
SO IS MARY POPPINS
```

Neither MALE nor FEMALE? In that program, what happens if the human says neither MALE nor FEMALE? For example, what happens if the human says SUPER-MALE or MACHO or NOT SURE or BOTH or YES? In those cases, the program doesn't tell the computer how to reply, so the computer will make no reply at all.

Let's improve the program, so that if the human says neither MALE nor FEMALE the computer will reply—

```
PLEASE SAY MALE OR FEMALE
ARE YOU MALE OR FEMALE?
```

and force the human to answer the question correctly. To do that, add this line—

```
40 PRINT "PLEASE SAY MALE OR FEMALE": GO TO 10
```

and put END at the end of lines 20 and 30, so that the program looks like this:

```
10 INPUT "ARE YOU MALE OR FEMALE";A$
20 IF A$="MALE" THEN PRINT "SO IS FRANKENSTEIN": END
30 IF A$="FEMALE" THEN PRINT "SO IS MARY POPPINS": END
40 PRINT "PLEASE SAY MALE OR FEMALE": GO TO 10
```

Line 10 makes the computer ask "ARE YOU MALE OR FEMALE?" and wait for the human's answer, which is called A$. If the human's answer is MALE, line 20 makes the computer print SO IS FRANKENSTEIN and then end. If the human's answer is FEMALE, line 30 makes the computer print SO IS MARY POPPINS and then end. If the human's answer is neither MALE nor FEMALE, the computer skips over lines 20 and 30, so it comes to line 40, which makes it print PLEASE SAY MALE OR FEMALE and then go back to line 10, which forces the human to answer the question again.

Here's a sample run:

```
RUN
ARE YOU MALE OR FEMALE? MALE
SO IS FRANKENSTEIN
```

Here's another:

```
RUN
ARE YOU MALE OR FEMALE? FEMALE
SO IS MARY POPPINS
```

Another:

```
RUN
ARE YOU MALE OR FEMALE? MACHO
PLEASE SAY MALE OR FEMALE
ARE YOU MALE OR FEMALE? MALE
SO IS FRANKENSTEIN
```

Another:

```
RUN
ARE YOU MALE OR FEMALE? SUPER-MALE
PLEASE SAY MALE OR FEMALE
ARE YOU MALE OR FEMALE? NONE OF YOUR BUSINESS
PLEASE SAY MALE OR FEMALE
ARE YOU MALE OR FEMALE? MAIL
PLEASE SAY MALE OR FEMALE
ARE YOU MALE OR FEMALE? MALE
SO IS FRANKENSTEIN
```

In that program, if the human makes a typing error, and answers neither MALE nor FEMALE, the computer arrives at line 40, which makes the computer gripe and tell the human to try again. So the purpose of line 40 is to react to errors. Line 40 is called an *error-handling routine* or an *error trap*. That's because an error's like a vicious monster, and line 40's purpose is to trap it.

Do you like Mary Poppins? Let's extend the conversation. If the human says FEMALE, let's make the computer say SO IS MARY POPPINS and then ask "DO YOU LIKE HER?" If the human says YES, let's make the computer say:

```
I LIKE HER TOO--SHE IS MY MOTHER
```

If the human says NO, let's make the computer say:

```
NEITHER DO I--SHE STILL OWES ME A DIME
```

If the human says neither YES nor NO, let's make the computer say:

```
PLEASE SAY YES OR NO
DO YOU LIKE HER?
```

Here's the program:

```
10 INPUT "ARE YOU MALE OR FEMALE";A$
20 IF A$="MALE" THEN PRINT "SO IS FRANKENSTEIN": END
30 IF A$="FEMALE" THEN PRINT "SO IS MARY POPPINS": GO TO 100
40 PRINT "PLEASE SAY MALE OR FEMALE": GO TO 10

100 INPUT "DO YOU LIKE HER";B$
110 IF B$="YES" THEN PRINT "I LIKE HER TOO--SHE IS MY MOTHER": END
120 IF B$="NO" THEN PRINT "NEITHER DO I--SHE STILL OWES ME A DIME": END
130 PRINT "PLEASE SAY YES OR NO": GO TO 100
```

Line 30 says: if the human's answer is FEMALE, print SO IS MARY POPPINS and then go to line 100, which asks "DO YOU LIKE HER?" Lines 110 and 120 make the computer react to the human's opinion of Mary Poppins. Line 130 is like line 40: it's an error trap.

The computer is like a human; it wants to make new friends. This program makes the computer show its true feelings:

```
10 INPUT "ARE YOU MY FRIEND";A$
20 IF A$="YES" THEN PRINT "THAT'S SWELL": END
30 IF A$="NO" THEN PRINT "GO JUMP IN A LAKE": END
40 PRINT "PLEASE SAY YES OR NO": GO TO 10
```

When you type RUN, the computer asks "ARE YOU MY FRIEND?" If you say YES, the computer says THAT'S SWELL. If you say NO, the computer says GO JUMP IN A LAKE.

The most inventive programmers are kids. This program was written by a girl in the sixth grade:

```
10 INPUT "CAN I COME OVER TO YOUR HOUSE TO WATCH T.V.";A$
20 IF A$="YES" THEN PRINT "THANKS.  I'LL BE THERE AT 5 P.M.": END
30 IF A$="NO" THEN PRINT "HUMPH!  YOUR FEET SMELL, ANYWAY.": END
40 PRINT "PLEASE SAY YES OR NO": GO TO 10
```

When you type RUN, the computer asks to watch your television. If you say YES, the computer promises to come to your house at 5. If you refuse, the computer insults your feet.

Another sixth-grade girl wrote this program, to test your honesty:

```
10 PRINT "FKGJDFGKJ*#K$JSLF*/#$()$&(IKJNHBGD52:?./KSDJK$E(EF$#/JIK(*"
20 PRINT "FASDFJKL:JFRFVFJUNJI*&()JNE$#SKI#(!SERF HHW NNWAZ MAME !!!"
30 PRINT "ZBB%%%%%##)))))FESDFJK DSFE N.D.JJUJASD EHWLKD******"
40 INPUT "DO YOU UNDERSTAND WHAT I SAID";A$
50 IF A$="NO" THEN PRINT "SORRY TO HAVE BOTHERED YOU": END
60 IF A$="YES" THEN GO TO 100
70 PRINT "PLEASE SAY YES OR NO": GO TO 10

100 PRINT "SSSFJSLFKDJFL++++4567345677839XSDWFEGF/#$&**()---==!!ZZXX"
110 PRINT "###EDFHTG NVFDF MKJKF ==+--*$&% #RHFS SESD DOPEKKKNJGFD DSBS"
120 INPUT "OKAY, WHAT DID I SAY";B$
130 PRINT "YOU ARE A LIAR, A LIAR, A BIG FAT LIAR!"
```

When you type RUN, lines 10-30 print nonsense. Then the computer asks whether you understand that stuff. *If you're honest* and answer NO, the computer will apologize. But *if you pretend that you understand the nonsense*, and answer YES, the computer will print more nonsense, challenge you to translate it, wait for you to fake a translation, and then scold you for lying.

A Daddy wrote a program for his five-year-old son, John. When John types RUN, the computer asks "WHAT'S 2 AND 2?" If John answers 4, the computer says NO, 2 AND 2 IS 22. If he runs the program again and answers 22, the computer says NO, 2 AND 2 IS 4. No matter how many times he runs the program and how he answers the question, the computer says he's wrong. But when Daddy runs the program, the computer replies, YES, DADDY IS ALWAYS RIGHT. Here's how Daddy programmed the computer:

```
10 INPUT "WHAT'S YOUR NAME";N$
20 INPUT "WHAT'S 2 AND 2";A
30 IF N$="DADDY" THEN PRINT "YES, DADDY IS ALWAYS RIGHT": END
40 IF A=4 THEN PRINT "NO, 2 AND 2 IS 22": END
50 PRINT "NO, 2 AND 2 IS 4"
```

FANCY CONDITIONS

You can make the IF clause very fancy:

IF clause	Meaning
IF A$="MALE"	If A$ is "MALE"
IF A=4	If A is 4
IF A<4	If A is less than 4
IF A>4	If A is greater than 4
IF A<=4	If A is less than or equal to 4
IF A>=4	If A is greater than or equal to 4
IF A<>4	If A is not 4
IF A$<"MALE"	If A$ is a word that comes before "MALE" in the dictionary
IF A$>"MALE"	If A$ is a word that comes after "MALE" is the dictionary

The word OR. The computer understands the word OR. For example, here's how to say, "If X is either 7 or 8, print the word WONDERFUL":

```
IF X=7 OR X=8 THEN PRINT "WONDERFUL"
```

That example is composed of two conditions: the first condition is "X=7"; the second condition is "X=8". Those two conditions combine, to form "X=7 OR X=8", which is called a *compound condition*.

If you use the word OR, put it between two conditions.

Right: `IF X=7 OR X=8 THEN PRINT "WONDERFUL"` *("X=7" and "X=8" are conditions.)*
Wrong: `IF X=7 OR 8 THEN PRINT "WONDERFUL"` *("8" is not a condition.)*

The word AND. The computer understands the word AND. Here's how to say, "If P is more than 5 but less than 10, print TUNA FISH":

```
IF P>5 AND P<10 THEN PRINT "TUNA FISH"
```

Here's how to say, "If S is at least 60 but less than 65, print YOU ALMOST FAILED":

```
IF S>=60 AND S<65 THEN PRINT "YOU ALMOST FAILED"
```

Here's how to say, "If N is a number from 1 to 10, print "THAT'S GOOD":

```
IF N>=1 AND N<=10 THEN PRINT "THAT'S GOOD"
```

ELSE

Here's how to say, "If A is less than 18, print MINOR; but if A is *not* less than 18, print ADULT":

```
IF A<18 THEN PRINT "MINOR" ELSE PRINT "ADULT"
```

That line says to either PRINT "MINOR" or ELSE PRINT "ADULT". If A is less than 18, the computer will PRINT "MINOR"; otherwise, the computer will PRINT "ADULT".

Here's how to say, "If A is less than 18, print MINOR and print YOUNG; if A is *not* less than 18, print ADULT and print OLD":

```
IF A<18 THEN PRINT "MINOR": PRINT "YOUNG" ELSE PRINT "ADULT": PRINT "OLD"
```

Primitive computers don't understand the word ELSE. To find out whether your computer is primitive, check the appendix.

REMINDERS

Whenever you're confused, type this command:

```
LIST
```

To make the computer do your program again and again, automatically, add an extra line at the bottom of your program. That extra line should tell the computer to GO TO the top line.

Here are two reasons why the computer might print too much:

You forgot to insert the word END into your program.
You forgot to type NEW; so the computer is reprinting part of the previous program.

You must put quotation marks around each string, and a dollar sign after each string variable:

Right	Wrong	Wrong
A$="JERK"	A$=JERK	A="JERK"

To say "not equal to", say "less than or greater than", like this: <>.

Here's the "proper" way to write an IF statement. . . . The IF statement should contain the word THEN. Do *not* put a comma before THEN. In an IF clause, the equal sign should come last:

Right	Wrong
<=	=<
>=	=>

DISKS

While you're working with a program, the computer keeps the program in its *random-access memory (RAM)*. In a typical microcomputer, the RAM can handle only one program at a time. If you want the computer to remember *several* programs, you can put one program in the RAM but must put the remaining programs elsewhere: put them on a disk or on a tape.

Here's how to use a disk. (I'll explain tape later.)

Before typing your program, make sure the computer and the disk have both been properly prepared. To find out how to prepare *your* computer and disk, read the appendix.

Then type your program. For example, try typing this program:

```
NEW
10 PRINT "MY DAD"
20 PRINT "IS GLAD"
```

Underneath the program, try typing this:

```
SAVE "JOE"
```

That makes the computer copy the program from the RAM to the disk, and name the program JOE.

Copying from the RAM to the disk is called *saving* the program. SAVE "JOE" means: save the program and name it "JOE".

Altogether, here's what you typed:

```
NEW
10 PRINT "MY DAD"
20 PRINT "IS GLAD"
SAVE "JOE"
```

(On *your* computer, the procedure might be slightly different: check the appendix.)

To make sure that the program did indeed get onto the disk, type this word:

```
FILES
```

The computer will print the names of all the programs on the disk. And one of the names it will print is JOE.

The program is now in *two* places: the original version of the program is still in the RAM, and a copy of the program is on the disk; the copy is called JOE.

If you type NEW, you'll erase the RAM; and so that program won't be in the RAM anymore. But even though the original version of the program isn't in the RAM anymore, the *copy* (which is called JOE) still remains on the disk. To prove that JOE is still on the disk, type the word FILES again; the computer will say again that JOE is one of the programs on the disk.

How to choose a name. If you don't like the name JOE, choose a different name instead. For example you can create a program named SUE, by typing a program and then saying:

```
SAVE "SUE"
```

The name of your program can be "JOE" or "SUE" or any other brief string of letters and digits. For example, it can be "LOVER" or "POEM4U".

Pick a name that reminds you of the program's purpose. For example, if the program prints a bill to a customer, call the program "BILL"; if the program plays chess, call the program "CHESS"; if the program gives a quiz, call the program "QUIZ"; if the program tutors a human about the elements of sex, call the program "SEX"; if the program tutors a human about advanced sex, call the program "SEX2".

Loading from disk. You've seen that the word SAVE makes the computer copy a program from the RAM to the disk. The opposite of the word SAVE is the word LOAD: it makes the computer copy a program from the *disk* to the *RAM*.

For example, if you want the computer to copy JOE from the disk to the RAM, type this:

```
LOAD "JOE"
```

It tells the computer to load JOE into the RAM (from the disk). After typing that, the program is in the RAM again. To prove the program's in the RAM, type the word LIST, so that the computer will LIST the program that's in the RAM. Or, if you prefer, type the word RUN, which makes the computer RUN the program that's in the RAM.

So if JOE's on a disk, you can run it by first copying it to the RAM and then typing the word RUN, like this:

```
LOAD "JOE"
RUN
```

Instead of typing that pair of lines, you can type this single line, which is a short-cut:

```
RUN "JOE"
```

That makes the computer automatically load JOE (from the disk to the RAM) and run it.

Editing the disk. Suppose you don't like the version of JOE that's on the disk. Here's how to revise JOE.

First, copy JOE from the disk to the RAM, by saying:

```
LOAD "JOE"
```

Then examine the program, by typing:

```
LIST
```

Then revise the program, by retyping some of the program's lines, or by adding new ones. When you've finished revising the version of the program that's in the RAM, you must copy the revisions from the RAM to the disk, by saying again:

```
SAVE "JOE"
```

That command makes the computer copy the revised version of JOE onto the disk (and it also erases the previous version of JOE from the disk).

If JOE is on the disk and you want to change its name to FRED, type this:

```
RENAME "JOE" TO "FRED"
```

Protect yourself. Suppose you're creating a program named TOM. Suppose TOM is so long that it takes you several hours to type. Wouldn't you be depressed if, after several hours of typing, your town suddenly had a blackout, and the computer forgot everything you'd typed?

To protect yourself against such calamities, type SAVE "TOM" every fifteen minutes. That way, if your town has a blackout, you'll lose only the few minutes of your work; the rest of your work will have already been saved on a disk.

Typing SAVE "TOM" every fifteen minutes protects you against *several* kinds of calamities. It protects you against blackouts; it protects you against "computer malfunction"; and it protects you against any careless error you might make.

Each time you say SAVE "TOM", the computer makes a copy of the program that's in the RAM, puts the copy onto a disk, and saves that copy permanently—or at least until you type SAVE "TOM" again fifteen minutes later. Even if your RAM is erased (by a blackout or computer malfunction or your carelessness), TOM will still be on the disk.

Space on the disk. If you invent many programs and put them on the disk, the disk will someday become full. If you try to put even more programs onto the disk, and there isn't any more room on the disk for them, the computer will gripe, by printing a message such as:

```
DISK FULL
```

Someday, you'll decide that you no longer have any use for JOE, and that JOE's just wasting space on your disk. To erase JOE from the disk, type this:

```
KILL "JOE"
```

TAPES

A tape resembles a disk but is cheaper and slower.

Here's how to put a program onto a cassette tape. First, type the program. Then put a tape into the tape recorder; to do that, you might have to press the recorder's EJECT button. Rewind the tape, by pressing the recorder's REWIND button. When the tape is rewound, press the recorder's STOP button. If the recorder has a counter, press the counter's button, so the counter becomes zero.

Then press the recorder's FAST FORWARD button, until you get to a part of the tape that's blank and good. (The very beginning of the tape is bad. Wait until the counter gets to at least 10.) Then press the STOP button.

If the recorder has a counter, notice the counter's number, and write it onto the tape's label, by using a pencil.

To find out what to do next, read the appendix, which explains the peculiarities of *your* computer!

DEBUGGING

If you write your own program and run it, probably it won't work. Your first reaction will be to blame the computer; don't! The probability is 99.99% that the fault is yours. Your program contains an error. An error is called a *bug*. Your next task is to *debug* the program, which means get the bugs out.

Bugs are common; top-notch programmers make errors all the time. If you write a program that works perfectly on the first run and doesn't need debugging, it's called a *gold-star program*, and means you should have tried writing a harder one instead!

It's easy to write a program that's almost completely correct, but hard to find the little bug that's fouling it up. Most of the time you spend at the computer will be devoted to debugging.

Debugging can be fun. Hunting for the bug is like going on a treasure hunt—or solving a murder mystery. Pretend you're Sherlock Holmes. Your mission: to find the bug, and squash it!

To find the bug, use three techniques:

Inspect the program.
Trace the computer's thinking.
Shorten the program.

Here are the details. . . .

Inspect the program. Take a good, hard look at the program. If you stare hard enough, maybe you'll see the bug.

Ask the computer to help you. Make the computer print the entire program. To do that, type:

```
LIST
```

Usually, the bug will turn out to be just a typing error. Maybe you:

typed the letter O instead of a zero? or typed zero instead of the letter O?
typed I instead of 1? or typed 1 instead of I?
pressed the SHIFT key when you weren't supposed to? or forgot *to press it?*
typed an extra letter? or omitted *a letter?*
typed a line you thought you hadn't? or omitted *a line?*

Trace the computer's thinking. If you've inspected the program thoroughly, and *still* haven't found the bug, the next step is to *trace* the computer's thinking. Pretend *you* are the computer. Do what your program says. Do you find yourself printing the same wrong answers the computer printed? If so, why?

To help your analysis, make the computer print *everything* it's thinking, while it's running your program. For example, suppose your program contains lines 10, 20, 30, and 40, and uses the variables B, C, and X$. Insert these lines into your program:

```
15 PRINT "I'M AT LINE 15.  THE VALUES ARE";B;C;X$
25 PRINT "I'M AT LINE 25.  THE VALUES ARE";B;C;X$
35 PRINT "I'M AT LINE 35.  THE VALUES ARE";B;C;X$
45 PRINT "I'M AT LINE 45.  THE VALUES ARE";B;C;X$
```

Then type the word RUN. The computer will run the program again; but lines 15, 25, 35, and 45 make the computer print everything it's thinking. Check what the computer prints. If the computer prints what you expect in lines 15 and 25, but prints strange values in line 35 (or doesn't even get to line 35), you know the bug occurs before line 35 but after line 25; so the bug must be in line 30.

If your program contains hundreds of lines, you might not have the patience to insert lines 15, 25, 35, 45, 55, 65, 75, etc. Here's a short-cut. . . .

Halfway down your program, insert a line that says to print all the values. Then run your program. If the line you inserted prints the correct values, you know the bug lies underneath that line; but if the line prints *wrong* values (or if the computer never reaches that line), you know the bug lies *above* that line. In either case, you know which half of your program contains the bug. In that half of the program, insert more lines, until you finally zero in on the line that contains the bug.

Shorten the program. When all else fails, shorten the program.

Hunting for a bug in a program is like hunting for a needle in a haystack: the job is easier if the haystack is smaller. So make your program shorter: delete the last half of your program. Then run the shortened version. That way, you'll find out whether the first half of your program is working the way it's supposed to. When you've perfected the first half of your program, tack the second half back on.

Does your program contain a statement whose meaning you're not completely sure of? Check the meaning, by reading a book or asking a friend; or write a tiny experimental program that contains the statement, and see what happens when you type RUN.

Hint: before you shorten your program (or write tiny experimental ones), SAVE the original version, even though it contains a bug. After you've played with the shorter versions, retrieve the original and fix it.

The easiest way to write a long, correct program is to write a short program first, debug it, then add a few more lines, debug them, add a few more lines, debug them, etc. In other words, start with a small program, perfect it, and then gradually add perfected extras, so you *gradually* build a perfected masterpiece. If you try to compose a long program all at once—instead of building it from perfected pieces—you'll have nothing more than a master*mess*—full of bugs.

Moral: to build a large masterpiece, start will a *small* masterpiece. Putting it another way: to build a skyscraper, begin by laying a good foundation; double-check the foundation, before you start adding the walls and the roof.

EDITING

Suppose you're typing a program and make a mistake. Here's how to correct the mistake quickly, by using the Walter keyboard. (Check the appendix, to find out how *your* computer differs from the Walter keyboard.)

First, make sure the line you want to correct is on the screen. If it's not on the screen yet, put it onto the screen by using the word LIST or EDIT. For example, to put line 30 onto the screen, type—

```
LIST 30
```

or type:

```
EDIT 30
```

On the keyboard, you'll see a key that shows an arrow pointing to the left; it's called the *left-arrow key*. You'll also see a key that shows an arrow pointing to the right; it's called the *right-arrow key*. You'll also see an *up-arrow key* and a *down-arrow key*.

On the computer's screen, you'll see a little blinking rectangle, called the *cursor*. Move it to the part of the line you want to correct, by pressing the arrow keys. (To move the cursor to the left, press the left-arrow key; to move the cursor to the right, press the right-arrow key; to move the cursor up, press the up-arrow key; to move the cursor down, press the down-arrow key.)

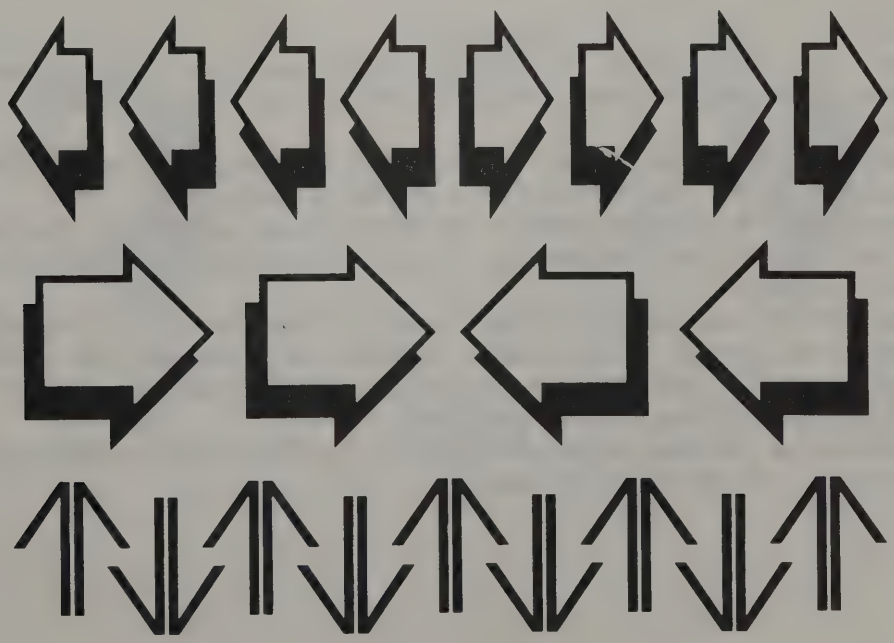
If you typed the word LIST, you must press the up-arrow key to move the cursor up to the line you want to correct. If you typed the word EDIT instead, you don't need to press the up-arrow key; the word EDIT puts the cursor on the correct line automatically.

When you've moved the cursor to the part of the line you want to correct, make your corrections. To delete a character, move the cursor to that character, then press the DELETE key. To replace a character, move the cursor to the character you want to replace, then type the new character you want instead. To insert extra characters in the middle of the line, move the cursor to the place where you want the extra characters to begin appearing, press the INSERT key, then type the extra characters.

When you've finished correcting the line, press the ENTER key, which tells the computer to take the corrections seriously.

After pressing the ENTER key, you might like to move the cursor to the bottom of the screen. If you'd like to do that, press the down-arrow key several times.

That method of editing works on *most* computers. To find out how to edit easily on *your* computer, read the appendix.



RENUMBERING

Suppose you've written a sloppy program, like this:

```
37 PRINT "SNARL"
68 PRINT "SMIRK"
91 GO TO 37
```

If you type—

```
RENUM
LIST
```

the computer will renumber your program, so that the lines are numbered 10, 20, and 30. The computer will say:

```
10 PRINT "SNARL"
20 PRINT "SMIRK"
30 GO TO 10
```

Notice that the computer even changed the "GO TO 37", so that it became "GO TO 10"!

If you type—

```
RENUM 500
LIST
```

the computer will renumber your program so that it begins at line 500. The computer will say:

```
500 PRINT "SNARL"
510 PRINT "SMIRK"
520 GO TO 500
```

Suppose you wrote a program that contained a line numbered 83, and you'd like that line to be called "line 500" instead. Here's how to start renumbering at line 83, so that line 83 becomes 500, and the next line becomes 600, and the line after *that* becomes 700, etc.:

```
RENUM 500,83,100
LIST
```

That command makes the computer renumber the program, so that 500 is the new number for line 83, and so that the lines after 500 go up by 100's.

The RENUM command works on *most* computers. To find out about *your* computer, read the appendix.

PAUSING

Magicians often say, "The hand is quicker than the eye." The computer's the ultimate magician: the computer can print information on the screen much faster than you can read it.

When the computer is printing too fast for you to read, tap the PAUSE key. The computer will pause, to let you read what's on the screen.

When you've finished reading what's on the screen, and want the computer to stop pausing, tap the PAUSE key again. Then the computer will continue printing rapidly, where it left off.

If your eyes are as slow as mine, you'll need to use the PAUSE key often! You'll want the computer to pause, when you're running a program that contains many PRINT statements, or if you're running a program that contains a PRINT statement in a loop, or if you're LISTing a long program.

The PAUSE key is on the Walter keyboard. To find out whether it's on *your* computer's keyboard, check the appendix.

APOSTROPHE

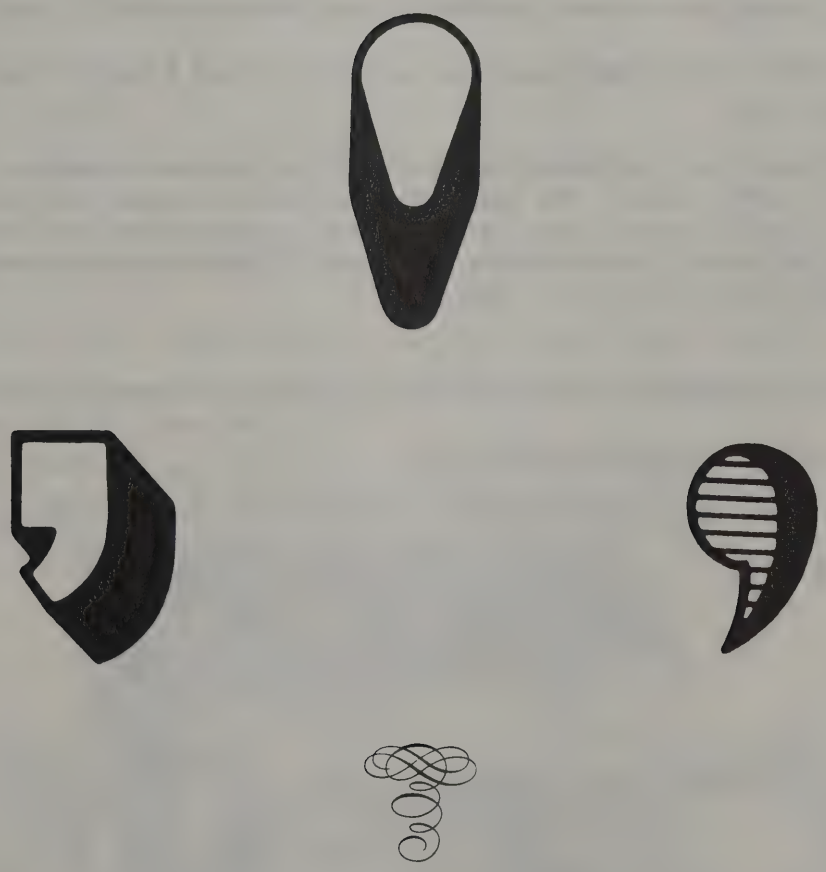
Occasionally jot a note, to remind yourself what your program does and what the variables stand for. Slip the note into your program, by putting an apostrophe before it:

```
10 'THIS PROGRAM IS ANOTHER DUMB EXAMPLE
20 'WRITTEN BY RUSSY-POO, A STUPID JERK
30 'IT WAS WRITTEN ON HALLOWEEN, UNDER A FULL MOON, SHADED BY HIS FANGS
40 C=40' BECAUSE RUSS HAS 40 COMPUTERS
50 H=23' BECAUSE 23 OF HIS COMPUTERS ARE HAUNTED
60 PRINT C-H' THAT IS HOW MANY COMPUTERS ARE UNHAUNTED AND SAFE FOR KIDS
```

When you type RUN, the computer ignores everything that's to the right of an apostrophe. So the computer ignores lines 10, 20, and 30; in lines 40 and 50 the computer ignores the "because . . . "; in line 60 the computer ignores the comments about being unhaunted. Since C is 40, and H is 23, line 60 makes the computer print:

```
 17
```

Everything to the right of an apostrophe is called a *comment* (or *remark*). While the computer runs the program, it ignores the comments. But whenever you say LIST, the computer will LIST the entire program, including even the comments. Although the comments appear in the LIST, they don't affect the RUN.



WOW! YOU FINISHED CHAPTER 4: PERSONAL PROGRAMMING!
TURN THE PAGE FOR CHAPTER 5: POWERFUL PROGRAMMING.

DATA . . . READ

LISTS OF DATA

Let's make the computer print this message:

```
I LOVE MEAT
I LOVE POTATOES
I LOVE LETTUCE
I LOVE TOMATOES
I LOVE BUTTER
I LOVE CHEESE
I LOVE ONIONS
I LOVE PEAS
```

That message concerns this list of food: MEAT, POTATOES, LETTUCE, TOMATOES, BUTTER, CHEESE, ONIONS, PEAS. That list doesn't change: the computer continues to love those foods throughout the entire program.

A list that doesn't change is called *data*. So in the message about food, the *data* is MEAT, POTATOES, LETTUCE, TOMATOES, BUTTER, CHEESE, ONIONS, PEAS.

Whenever a problem involves data, put the data at the top of the program, like this:

```
10 DATA MEAT,POTATOES,LETTUCE,TOMATOES,BUTTER,CHEESE,ONIONS,PEAS
```

You must tell the computer to READ the DATA:

```
20 READ A$
```

Line 20 makes the computer read the first datum (which is MEAT), and call it A$. So A$ is MEAT.

Since A$ is MEAT, this line makes the computer print I LOVE MEAT:

```
30 PRINT "I LOVE ";A$
```

Hooray! We made the computer handle the first datum correctly: we made the computer print I LOVE MEAT. To make the computer handle the rest of the data (POTATOES, LETTUCE, etc.), tell the computer to READ the rest of the data: tell the computer to GO back TO the READ statement. Since the READ statement is line 20, tell the computer to GO TO 20, like this:

```
40 GO TO 20
```

Line 40 makes the computer GO back TO line 20, which reads the next datum (POTATOES).

Altogether, the program looks like this:

```
10 DATA MEAT,POTATOES,LETTUCE,TOMATOES,BUTTER,CHEESE,ONIONS,PEAS
20 READ A$
30 PRINT "I LOVE ";A$
40 GO TO 20
```

Lines 20-40 form a *loop*. (Like most loops, the loop's bottom line says GO TO. Since the loop's top line says READ, the loop is called a *READ loop*.) The computer goes round and round the loop. Each time the computer comes to line 20, it reads another datum: the first time it comes to line 20, it reads MEAT; the next time, it reads POTATOES; the next time, it reads LETTUCE; the next time, it reads TOMATOES; etc. Line 30 makes the computer PRINT what it's read. Altogether, the computer will print:

```
I LOVE MEAT
I LOVE POTATOES
I LOVE LETTUCE
I LOVE TOMATOES
I LOVE BUTTER
I LOVE CHEESE
I LOVE ONIONS
I LOVE PEAS
```

After the computer prints I LOVE PEAS, it comes to line 40 again, which makes it go back to line 20 again, so the computer tries to read more data again; but no more data remains! So the computer says:

```
OUT OF DATA
```

Then the computer stops.

So the last three lines the computer prints are:

```
I LOVE ONIONS
I LOVE PEAS
OUT OF DATA
```

Instead of saying OUT OF DATA, let's make the computer say "I LOVE ALL THOSE FOODS", so that the last three lines look like this:

```
I LOVE ONIONS
I LOVE PEAS
I LOVE ALL THOSE FOODS
```

Here's how. . . . Underneath the data, say DATA END:

```
15 DATA END
```

When the computer reads the DATA END, make the computer say I LOVE ALL THOSE FOODS and end:

```
20 READ A$: IF A$="END" THEN PRINT "I LOVE ALL THOSE FOODS": END
```

The DATA END underneath the data is called the *end mark*, because it marks the data's end. The routine that says—

```
IF A$="END" THEN PRINT "I LOVE ALL THOSE FOODS": END
```

is called the *end routine*, because the computer does that routine at the end.

That program prints one copy of the computer's favorite foods. If you want the computer to print *many* copies, change line 20 to this:

```
20 READ A$: IF A$="END" THEN PRINT "I LOVE ALL THOSE FOODS": RESTORE: GO TO 20
```

The word RESTORE tells the computer to go back to the beginning of the data. The computer will print:

```
I LOVE MEAT
I LOVE POTATOES
I LOVE LETTUCE
I LOVE TOMATOES
I LOVE BUTTER
I LOVE CHEESE
I LOVE ONIONS
I LOVE PEAS
I LOVE ALL THOSE FOODS
I LOVE MEAT
I LOVE POTATOES
I LOVE LETTUCE
I LOVE TOMATOES
I LOVE BUTTER
I LOVE CHEESE
I LOVE ONIONS
I LOVE PEAS
I LOVE ALL THOSE FOODS
I LOVE MEAT
I LOVE POTATOES
```

etc.

The computer will print copies of what it likes again and again, forever, unless you abort the program.

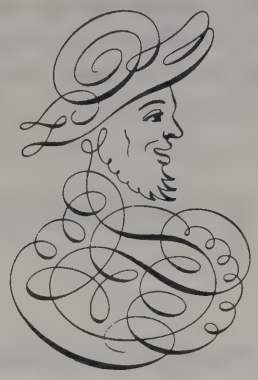
Most practical computer programs involve data (a list that doesn't change). Your job, as a programmer, is to notice what the data is, and to begin your program by saying DATA. After you type the data, the next statement in your program should say READ. Farther down in your program, you should say GO TO, so that you create a READ loop. Your program consists of two parts: the DATA, and the READ loop.

You must put quotation marks around a data string, if the string contains a comma or a colon. For example, if one of the foods is "HOT, JUICY, BIG, THICK STEAKS" (which contains commas), you must put quotation marks around it.

HENRY THE EIGHTH

Let's make the computer print this nursery rhyme:

I love ice cream
I love red
I love ocean
I love bed
I love tall grass
I love to wed



I love candles
I love divorce
I love my kingdom
I love my horse
I love you
Of course, of course,
For I am Henry the Eighth!

If you own a jump rope, you can have fun: try to recite that poem while skipping rope!

This program makes the computer recite the poem repeatedly:

```
10 DATA ICE CREAM,RED,OCEAN,BED,TALL GRASS,TO WED
11 DATA CANDLES,DIVORCE,MY KINGDOM,MY HORSE,YOU
15 DATA END
20 READ A$: IF A$="END" THEN PRINT "OF COURSE, OF COURSE,": PRINT "FOR I AM HENR
Y THE EIGHTH!": PRINT: RESTORE: GO TO 20
30 PRINT "I LOVE ";A$
35 IF A$="TO WED" THEN PRINT
40 GO TO 20
```

Since the data's too long to fit on a single line, I've put part of the data in line 10 and the rest in line 11. Each line of data must begin with the word DATA. In each line, put comas between the items. Do *not* put a comma at the end of the line.

The program resembles the previous one. The only new line is 35, which makes the computer leave a blank line underneath "TO WED", to mark the bottom of the first verse.

PARTY

Let's throw a party! To make the party yummy, let's ask each guest to bring a kind of food that resembles the guest's name. For example, let's have Sal bring salad, Russ bring Russian dressing, Sue bring soup, Tom bring turkey, Winnie bring wine, Kay bring cake, and Al bring Alka-Seltzer.

Let's send all those people invitations, in this form:

```
DEAR _____________
     person's name

     WE'RE THROWING A PARTY AT THE SECRET CLUBHOUSE TOMORROW AT NOON!

PLEASE BRING ____
             food
```

To make the computer print all the invitations, begin by feeding the computer the DATA:

```
10 DATA SAL,SALAD,RUSS,RUSSIAN DRESSING,SUE,SOUP,TOM,TURKEY
20 DATA WINNIE,WINE,KAY,CAKE,AL,ALKA-SELTZER
```

The data comes in pairs: the first pair consists of SAL and SALAD. Tell the computer to READ each pair of DATA:

```
30 READ P$,F$
```

Line 30 makes the computer read the first pair of data; so P$ is the first person (SAL), and F$ is his food (SALAD).

These lines print the letter:

```
40 PRINT "DEAR ";P$
50 PRINT "     WE'RE THROWING A PARTY AT THE SECRET CLUBHOUSE TOMORROW AT NOON!"
60 PRINT "PLEASE BRING ";F$
```

At the end of the letter, leave two blank lines, to make room for your signature:

```
70 PRINT
80 PRINT
```

Then complete the READ loop, by making the computer go back to the beginning of the loop, to read the next pair:

```
90 GO TO 30
```

The computer will print a letter to each person, like this:

```
DEAR SAL
     WE'RE THROWING A PARTY AT THE SECRET CLUBHOUSE TOMORROW AT NOON!
PLEASE BRING SALAD


DEAR RUSS
     WE'RE THROWING A PARTY AT THE SECRET CLUBHOUSE TOMORROW AT NOON!
PLEASE BRING RUSSIAN DRESSING
```

etc.

After printing all the letters, the computer will say:

```
OUT OF DATA
```

Instead of saying OUT OF DATA, let's make the computer say:

```
I'VE FINISHED WRITING THE LETTERS
```

To do that, put END at the end of the data, *twice*—

```
25 DATA END,END
```

and say what to do when the computer reaches the END:

```
30 READ P$,F$: IF P$="END" THEN PRINT "I'VE FINISHED WRITING THE LETTERS": END
```

You need *two* ENDs at the end of the data, because the READ statement says to read two strings (P$ and F$).

DEBTS

Suppose these people owe you things:

Person	What the person owes
Bob	$537.29
Mike	a dime
Sue	2 golf balls
Harry	a steak dinner at Mario's
Mommy	a kiss

Let's remind those people of their debts, by writing them letters, in this form:

```
DEAR _____________
     person's name

     I JUST WANT TO REMIND YOU...

THAT YOU STILL OWE ME ____
                      debt
```

Begin with the DATA:

```
10 DATA BOB,$537.29,MIKE,A DIME,SUE,2 GOLF BALLS
20 DATA HARRY,A STEAK DINNER AT MARIO'S,MOMMY,A KISS
```

The data comes in pairs: the first pair consists of BOB and $537.29. Tell the computer to READ each pair of DATA:

```
30 READ P$,D$
```

Line 30 makes the computer read the first pair of data; so P$ is the first person (BOB), and D$ is his debt ($537.29).

Here's the rest of the program:

```
40 PRINT "DEAR ";P$
50 PRINT "     I JUST WANT TO REMIND YOU..."
60 PRINT "THAT YOU STILL OWE ME ";D$
70 PRINT
80 PRINT
90 GO TO 30
```

The computer will print a letter to each person, like this:

```
DEAR BOB
     I JUST WANT TO REMIND YOU...
THAT YOU STILL OWE ME $537.29


DEAR MIKE
     I JUST WANT TO REMIND YOU...
THAT YOU STILL OWE ME A DIME
```

etc.

After printing all the letters, the computer will say:

```
OUT OF DATA
```

To prevent the computer from saying OUT OF DATA, add line 25 and retype line 30, as follows:

```
25 DATA END,END
30 READ P$,D$: IF P$="END" THEN PRINT "I'VE FINISHED WRITING THE LETTERS": END
```

DIETS

Suppose you're running a diet clinic and get these results:

Person	Weight before	Weight after
Joe	273 pounds	219 pounds
Mary	412 pounds	371 pounds
Bill	241 pounds	173 pounds
Sam	309 pounds	198 pounds

Here's how to make the computer print a nice report. . . .

Begin by feeding it the DATA:

```
10 DATA JOE,273,219,MARY,412,371,BILL,241,173,SAM,309,198
```

The DATA comes in triplets: the first triplet consists of JOE, 273, and 219. Tell the computer to READ each triplet of DATA:

```
20 READ N$,B,A
```

That line makes the computer read the first triplet of data; so N$ is the first person's name (JOE), B is his weight before (273), and A is his weight after (219). These lines print the report about him:

```
30 PRINT N$;" WEIGHED";B;"POUNDS BEFORE ATTENDING THE DIET CLINIC"
40 PRINT "BUT WEIGHED ONLY";A;"POUNDS AFTERWARDS"
50 PRINT "THAT'S A LOSS OF";B-A;"POUNDS"
```

At the end of the report about him, leave a blank line:

```
60 PRINT
```

Then complete the READ loop, by making the computer go back to the loop's beginning:

```
70 GO TO 20
```

The computer will print:

```
JOE WEIGHED 273 POUNDS BEFORE ATTENDING THE DIET CLINIC
BUT WEIGHED ONLY 219 POUNDS AFTERWARDS
THAT'S A LOSS OF 54 POUNDS

MARY WEIGHED 412 POUNDS BEFORE ATTENDING THE DIET CLINIC
BUT WEIGHED ONLY 371 POUNDS AFTERWARDS
THAT'S A LOSS OF 41 POUNDS
```

etc.

At the end, the computer will say OUT OF DATA. Instead, let's make it say:

```
COME TO OUR DIET CLINIC!
```

To do that, put END and two zeros at the end of the data—

```
15 DATA END,0,0
```

and say what to do when the computer reaches the END:

```
20 READ N$,B,A: IF N$="END" THEN PRINT "COME TO OUR DIET CLINIC!": END
```

You need the two zeros after the END, because the READ statement says to read two numbers (B and A) after the string N$. If you omit the zeros, the computer will say OUT OF DATA. If you hate zeros, you can use other numbers instead; but most programmers prefer zeros.

FRENCH COLORS

Let's make the computer translate colors into French. For example, if the human says RED, we'll make the computer say the French equivalent, which is:

```
ROUGE
```

Altogether, a run will look like this:

```
RUN
WHICH COLOR INTERESTS YOU? RED
IN FRENCH, IT'S ROUGE
```

The program begins simply:

```
10 INPUT "WHICH COLOR INTERESTS YOU";C$
```

Next, we must make the computer translate the color into French. To do that, feed the computer this English-French dictionary:

English	French
white	blanc
yellow	jaune
orange	orange
red	rouge
green	vert
blue	bleu
brown	brun
black	noir

That dictionary becomes the data:

```
20 DATA WHITE,BLANC,YELLOW,JAUNE,ORANGE,ORANGE,RED,ROUGE
30 DATA GREEN,VERT,BLUE,BLEU,BROWN,BRUN,BLACK,NOIR
```

The data comes in pairs: the first pair consists of WHITE and BLANC. Tell the computer to READ each pair of DATA:

```
40 READ E$,F$
```

That line makes the computer read the first pair of data; so E$ is the first English color (WHITE), and F$ is its French equivalent (BLANC).

But that pair of data might be the *wrong pair*; for example, if the human requested RED, the human does *not* want the pair of WHITE and BLANC; instead, the human wants the pair of RED and ROUGE. So you must tell the computer: if the human's input doesn't match the English in the pair, go read another pair. Here's how to say that:

```
50 IF E$<>C$ THEN GO TO 40
```

That line says: if E$ (which is the English in the data) is not C$ (the color the human requested), go to 40 (which reads another pair).

Lines 40 and 50 form a loop; the computer goes round and round the loop, until it finds the pair of data that matches the human's request. Since the purpose of the loop is to search for data that matches, it's called a *search loop*.

After the computer has found the correct English-French pair, make the computer print the French:

```
60 PRINT "IN FRENCH, IT'S ";F$
```

Altogether, the program looks like this:

```
Ask the human→→→→→→→→→→→10 INPUT "WHICH COLOR INTERESTS YOU";C$
Use this dictionary→→→→→20 DATA WHITE,BLANC,YELLOW,JAUNE,ORANGE,ORANGE,RED,ROUGE
                          30 DATA GREEN,VERT,BLUE,BLEU,BROWN,BRUN,BLACK,NOIR
Look at the dictionary→→40 READ E$,F$
If not found, try again→50 IF E$<>C$ THEN GO TO 40
Print the French→→→→→→→→60 PRINT "IN FRENCH, IT'S ";F$
```

Here's a sample run:

```
RUN
WHICH COLOR INTERESTS YOU? RED
IN FRENCH, IT'S ROUGE
```

Here's another:

```
RUN
WHICH COLOR INTERESTS YOU? BROWN
IN FRENCH, IT'S BRUN
```

Here's another:

```
RUN
WHICH COLOR INTERESTS YOU? PINK
OUT OF DATA
```

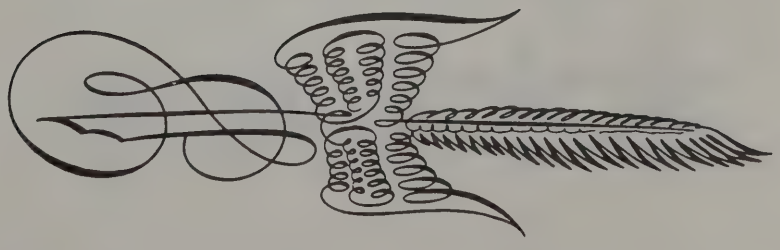
The computer says OUT OF DATA because it can't find PINK in the data. Instead of saying OUT OF DATA, it would be nicer to say I WASN'T TAUGHT THAT COLOR.

We want the computer to say I WASN'T TAUGHT THAT COLOR, when it reaches the end of the data. To do that, say END at the end of the data; and when the computer reaches the END, tell it to say I WASN'T TAUGHT THAT COLOR:

```
          10 INPUT "WHICH COLOR INTERESTS YOU";C$
          20 DATA WHITE,BLANC,YELLOW,JAUNE,ORANGE,ORANGE,RED,ROUGE
          30 DATA GREEN,VERT,BLUE,BLEU,BROWN,BRUN,BLACK,NOIR
Insert--->35 DATA END,END
Revised-->40 READ E$,F$: IF E$="END" THEN PRINT "I WASN'T TAUGHT THAT COLOR": END
          50 IF E$<>C$ THEN GO TO 40
          60 PRINT "IN FRENCH, IT'S ";F$
```

After line 60, the program just ends. Instead of letting the computer end, let's make it automatically rerun the program and translate another color. To do that, say GO TO and RESTORE:

```
  10 INPUT "WHICH COLOR INTERESTS YOU";C$
  20 DATA WHITE,BLANC,YELLOW,JAUNE,ORANGE,ORANGE,RED,ROUGE
  30 DATA GREEN,VERT,BLUE,BLEU,BROWN,BRUN,BLACK,NOIR,END,END
  35 DATA END,END
-->40 READ E$,F$: IF E$="END" THEN PRINT "I WASN'T TAUGHT THAT COLOR": GO TO 70
  50 IF E$<>C$ THEN GO TO 40
  60 PRINT "IN FRENCH, IT'S ";F$
-->70 RESTORE
-->80 GO TO 10
```



FOR . . . NEXT

COUNTING TO 100

Let's make the computer print every number from 1 to 100, like this:

```
1
2
3
4
5
6
7
etc.
100
```

To do that, type this line—

```
20      PRINT X
```

and also tell the computer that you want X to be every number from 1 to 100. To tell the computer that, say "FOR X = 1 TO 100", like this:

Insert this line→→
```
10 FOR X = 1 TO 100
20      PRINT X
```

Whenever you write a program that contains the word FOR, you must say NEXT. So your program should look like this:

```
10 FOR X = 1 TO 100
20      PRINT X
```
Insert this line→→
```
30 NEXT X
```

That program works; it makes the computer print every number from 1 to 100.

Here's how it works. . . . The computer begins at line 10, which says that you want X to be every number from 1 to 100. So X starts at 1. Then the computer comes to line 20, which says to print X; so the computer prints:

```
1
```

Then the computer comes to line 30, which says to do the same thing for the next X, and for the next X, and for the next X; so the computer prints 2, and 3, and 4, and so on, all the way up to 100.

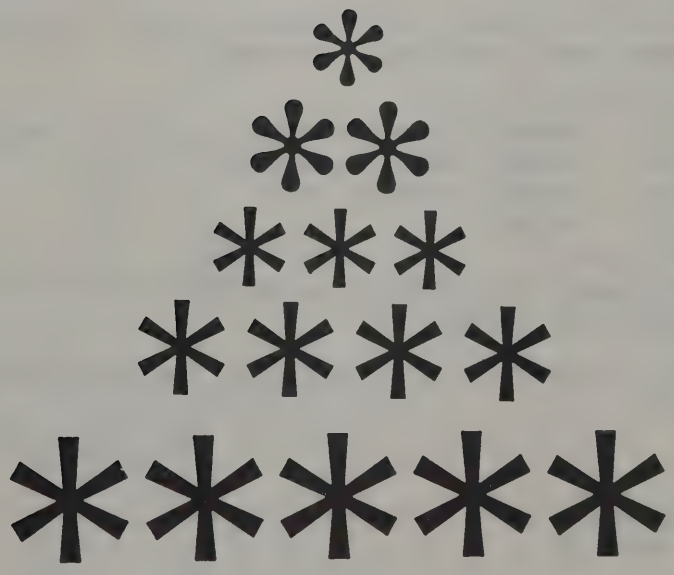
That program makes the computer print many numbers. That's because the computer does line 20 many times (once for each X).

The computer does line 20 many times, because line 20 is between the words FOR and NEXT: it's underneath FOR, and above NEXT. The computer repeats anything that's between the words FOR and NEXT.

Most programmers get in the habit of indenting everything that comes between the words FOR and NEXT; so in that program, I indented the statement that says PRINT X. To make the indentation, you can hit the SPACE bar repeatedly.

In line 30, if you're too lazy to type the X, you can omit it and just type:

```
30 NEXT
```



WHEN MEN MEET WOMEN

For a more advanced example, let's make the computer print these lyrics:

```
I SAW 2 MEN
MEET 2 WOMEN
TRA-LA-LA!

I SAW 3 MEN
MEET 3 WOMEN
TRA-LA-LA!

I SAW 4 MEN
MEET 4 WOMEN
TRA-LA-LA!

I SAW 5 MEN
MEET 5 WOMEN
TRA-LA-LA!

THEY ALL HAD A PARTY!
HA-HA-HA!
```

To do that, type these lines—

```
The first line of each verse→→→→→→→→→→20    PRINT "I SAW";X;"MEN"
The second line of each verse→→→→→→→→→30    PRINT "MEET";X;"WOMEN"
The third line of each verse→→→→→→→→→→40    PRINT "TRA-LA-LA!"
The blank line underneath each verse→→50    PRINT
```

and also tell the computer that you want X to be every number from 2 up to 5. Here's how:

```
Insert this line→→10 FOR X = 2 TO 5
                 20     PRINT "I SAW";X;"MEN"
                 30     PRINT "MEET";X;"WOMEN"
                 40     PRINT "TRA-LA-LA!"
                 50     PRINT
Insert this line→→60 NEXT
```

At the end of the song, print the closing couplet:

```
                 10 FOR X = 2 TO 5
                 20     PRINT "I SAW";X;"MEN"
                 30     PRINT "MEET";X;"WOMEN"
                 40     PRINT "TRA-LA-LA!"
                 50     PRINT
                 60 NEXT
Insert this line→→70 PRINT "THEY ALL HAD A PARTY!"
Insert this line→→80 PRINT "HA-HA-HA!"
```

That program works; it makes the computer print the entire song.

Here's an analysis:

```
                                          10 FOR X = 2 TO 5
The computer will do these lines repeatedly, {20     PRINT "I SAW";X;"MEN"
for X=2, X=3, X=4, and X=5.                  {30     PRINT "MEET";X;"WOMEN"
                                             {40     PRINT "TRA-LA-LA!"
                                             {50     PRINT
                                          60 NEXT
Then the computer will print this couplet once. {70 PRINT "THEY ALL HAD A PARTY!"
                                                {80 PRINT "HA-HA-HA!"
```

Since the computer does lines 20-50 repeatedly, those lines form a loop. Here's the general rule: the statements between FOR and NEXT form a loop. The computer goes round and round the loop, for X=2, X=3, X=4, and X=5. Altogether, it goes around the loop 4 times, which is a finite number. Therefore, the loop is *finite*.

If you don't like the letter X, choose a different letter. For example, you can choose the letter I:

```
→→10 FOR I = 2 TO 5
→→20     PRINT "I SAW";I;"MEN"
→→30     PRINT "MEET";I;"WOMEN"
  40     PRINT "TRA-LA-LA!"
  50     PRINT
  60 NEXT
  70 PRINT "THEY ALL HAD A PARTY!"
  80 PRINT "HA-HA-HA!"
```

Most programmers prefer the letter I, when using the word FOR. In other words, most programmers say "FOR I", instead of "FOR X". Saying "FOR I" is an "old tradition". I'll follow the tradition: in the rest of this book, I'll say "FOR I", except in situations where some other letter feels more natural.

SQUARES

To find the *square* of a number, multiply the number by itself. The square of 3 is "3 times 3", which is 9. The square of 4 is "4 times 4", which is 16.

Let's make the computer print the square of 3, 4, 5, etc., up to 100, like this:

```
THE SQUARE OF 3 IS 9
THE SQUARE OF 4 IS 16
THE SQUARE OF 5 IS 25
THE SQUARE OF 6 IS 36
THE SQUARE OF 7 IS 49
etc.
THE SQUARE OF 100 IS 10000
```

To do that, type this line—

```
20     PRINT "THE SQUARE OF";I;"IS";I*I
```

and also tell the computer that you want I to be every number from 3 up to 100. Here's the program:

```
10 FOR I = 3 TO 100
20     PRINT "THE SQUARE OF";I;"IS";I*I
30 NEXT
```

SECRET MEETING

This program prints 12 copies of the same message:

```
10 FOR I = 1 TO 12
20     PRINT "HUSH, HUSH!"
30     PRINT "  WE'RE HAVING A SECRET MEETING..."
40     PRINT "    IN THE COMPUTER ROOM..."
50     PRINT "      TONIGHT..."
60     PRINT "        AT 2 A.M."
70     PRINT "          WEAR A FUNNY HAT."
80     PRINT
90     PRINT
100 NEXT
```

Lines 80 and 90 leave blank lines at the end of each copy, for your signature.

MIDNIGHT

This program makes the computer count to midnight:

```
10 FOR I = 1 TO 12
20     PRINT I
30 NEXT
40 PRINT "MIDNIGHT"
```

The computer will print:

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
MIDNIGHT
```

Let's put a semicolon at the end of line 20:

```
  10 FOR I = 1 TO 12
→→20     PRINT I;
  30 NEXT
  40 PRINT "MIDNIGHT"
```

The semicolon makes the computer print each item on the same line, like this:

```
 1  2  3  4  5  6  7  8  9  10  11  12 MIDNIGHT
```

If you want the computer to press the ENTER key before MIDNIGHT, insert a PRINT line:

```
  10 FOR I = 1 TO 12
  20     PRINT I;
  30 NEXT
→→35 PRINT
  40 PRINT "MIDNIGHT"
```

Line 35 makes the computer press the ENTER key just before MIDNIGHT, so the computer will print MIDNIGHT on a separate line, like this:

```
 1  2  3  4  5  6  7  8  9  10  11  12
MIDNIGHT
```

In line 20, the semicolon means: do *not* press the ENTER key after I. Line 35 means: *do* press the ENTER key. So line 35 undoes line 20, and makes the computer press the ENTER key before MIDNIGHT.

Let's make the computer count to midnight 3 times, like this:

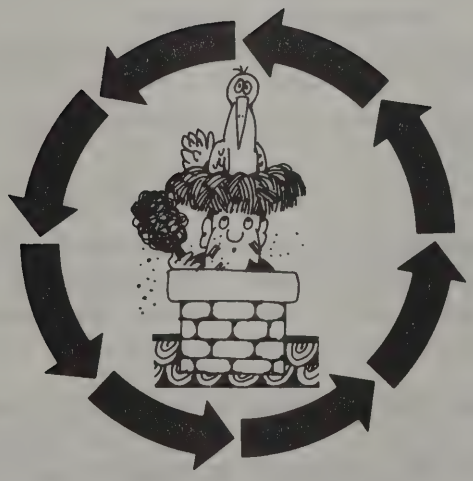
```
 1  2  3  4  5  6  7  8  9  10  11  12
MIDNIGHT
 1  2  3  4  5  6  7  8  9  10  11  12
MIDNIGHT
 1  2  3  4  5  6  7  8  9  10  11  12
MIDNIGHT
```

To do that, put the entire program between the words FOR and NEXT:

```
→→5 FOR A = 1 TO 3
  10     FOR I = 1 TO 12
  20         PRINT I;
  30     NEXT
  35     PRINT
  40     PRINT "MIDNIGHT"
→→50 NEXT
```

That version contains a loop inside a loop: the loop that says "FOR I" is inside the loop that says "FOR A". The A loop is called the *outer loop*; the I loop is called the *inner loop*. The inner loop's variable must differ from the outer loop's. Since we called the inner loop's variable "I", the outer loop's variable must *not* be called "I"; so I picked the letter A instead.

Often, programmers think of the outer loop as a bird's nest, and the inner loop as an egg *inside the nest*. So programmers say the inner loop is *nested in* the outer loop; the inner loop is a *nested loop*.



FAVORITE COLOR

This program plays a guessing game:

```
10 PRINT "I'LL GIVE YOU FIVE GUESSES...."
20 FOR I = 1 TO 5
30     INPUT "WHAT'S MY FAVORITE COLOR";G$
40     IF G$="PINK" THEN GO TO 100
50     PRINT "NO."
60 NEXT
70 PRINT "SORRY, YOUR FIVE GUESSES ARE UP! YOU LOSE."
80 END

100 PRINT "CONGRATULATIONS! YOU DISCOVERED MY FAVORITE COLOR."
110 PRINT "IT TOOK YOU";I;"GUESSES."
```

Line 10 warns the human that only five guesses are allowed. Line 20 makes the computer count from 1 to 5; to begin, I is 1. Line 30 asks the human to guess the computer's favorite color; the guess is called G$. If the guess is PINK, the computer jumps from line 40 to line 100, prints CONGRATULATIONS, and tells how many guesses the human took. But if the guess is *not* PINK, the computer proceeds from line 40 to line 50, prints NO, and goes on to the next guess. If the human guesses five times without success, the computer proceeds from line 60 to line 70 and prints SORRY...YOU LOSE.

For example, if the human's third guess is PINK, the computer prints:

```
CONGRATULATIONS! YOU DISCOVERED MY FAVORITE COLOR.
IT TOOK YOU 3 GUESSES.
```

If the human's very first guess is PINK, the computer prints—

```
CONGRATULATIONS! YOU DISCOVERED MY FAVORITE COLOR.
IT TOOK YOU 1 GUESSES.
```

—which is ungrammatical but understandable.

Lines 20-60 form a loop. Line 20 says the loop will normally be done five times. The line after the loop, line 70, is the loop's *normal exit*. But if the human happens to input PINK, the computer jumps out of the loop early, to line 100, which is the loop's *abnormal exit*.

FINITE PAUSE

Have you ever met someone who acts romantic, then suddenly says something cruel that breaks the romance?

Let's make the computer act that way. For example, let's make the computer begin by crooning a romantic message:

```
10 PRINT "YOUR LUSCIOUS LIPS SMELL LIKE THE FINEST WINE"
```

Then let's make the computer pause, to give the human a chance to read and admire that romantic message, and to make the human wait in suspense for the next outburst of computer emotion. Here's how to make the computer pause:

```
20 FOR I = 1 TO 7000: NEXT
```

That line makes the computer pause, while it counts up to 7000. That line makes the computer mumble to itself, "I is 1, I is 2, I is 3, I is 4, . . . " without printing anything on the screen. While the computer is secretly and silently mumbling to itself up to 7000, the human has a chance to read and admire the romantic message printed by line 10.

Finally, let's make the computer analyze the meaning of lips smelling like wine:

```
30 PRINT "YOU MUST BE DRUNK"
```

So altogether, that program makes the computer print YOUR LUSCIOUS LIPS SMELL LIKE THE FINEST WINE, then pause, then say YOU MUST BE DRUNK.

In that program, line 20 makes the computer pause, by making the computer silently mumble up to 7000. The typical computer mumbles 1000 times per second. Since line 20 says to mumble 7000 times, that line takes 7 seconds altogether, and so it makes the computer pause for 7 seconds.

If you want the computer to pause for 12 seconds instead of 7, retype line 20, so that to mumble 12000 times instead of 7000.

Although the *typical* computer mumbles 1000 times per second, *your* computer might mumble slightly faster or slower than that. Try the program on *your* computer, and notice how many seconds *your* computer pauses when doing 7000 mumbles. If your computer's typical, it pauses 7 seconds; if your computer mumbles quickly, it pauses *less* than 7 seconds; if your computer mumbles slowly, it pauses *more* than 7 seconds.

Warning: your computer mumbles more slowly in long programs than in short ones.

This program makes the computer print a famous joke:

```
10 PRINT "YOUR TEETH ARE LIKE STARS"
20 FOR I = 1 TO 7000: NEXT
30 PRINT "THEY COME OUT AT NIGHT"
```

Line 30 is the punch line. Line 20 makes the computer pause 7 seconds, before giving the punch line.

Experiment: invent your *own* joke, and make the computer pause before printing the punch line.

Eye test. This program makes the computer give you an eye test; the computer will test how fast you can read:

```
10 CLS
20 PRINT "IF YOU CAN READ THIS, YOU READ QUICKLY"
30 FOR I = 1 TO 250: NEXT
40 CLS
```

Line 10 makes the computer clear the screen, so that the entire screen is blank. Line 20 makes the computer print:

```
IF YOU CAN READ THIS, YOU READ QUICKLY
```

Line 30 makes the computer pause for a quarter of a second (since 250 is a quarter of 1000); while the computer is pausing, the message stays on the screen. Line 40 erases the screen, so the message stayed on the screen for only a quarter of a second. If you were able to read the message in a quarter of second, you have very quick eyes!

I can't read that quickly. Can you? Try it! To make the eye test easier, allow yourself longer than a quarter of a second to read the message, by changing the 250 to a larger number.

Get together with your friends, and see how quickly they can read. Flash different messages on the screen, by changing line 20. For example, try changing line 20 to this:

```
20 PRINT "MUMBLING MORONS MAKE MY MOM MISS MURDER MYSTERIES MONDAY MORNING"
```

If your friends can read all that in a quarter of a second, they probably belong in the Guinness Book of World Records.

To prevent your friends from cheating, make them close their eyes while you're typing the program. When you've finished typing the program, press the CLEAR key, so that the program becomes invisible. Then tell your friends to open their eyes and type RUN.

The FOR statement can be varied:

Statement	Meaning
`FOR I = 5 TO 17 STEP .1`	I will be every number from 5 to 17, counting by tenths. So I will be 5, then 5.1, then 5.2, etc., up to 17.
`FOR I = 5 TO 17 STEP 3`	I will be every third number from 5 to 17. So I will be 5, then 8, then 11, then 14, then 17.
`FOR I = 17 TO 5 STEP -3`	I will be every third number from 17 down to 5. So I will be 17, then 14, then 11, then 8, then 5.

To count down, you must use the word STEP. To count from 17 down to 5, give this instruction:

```
FOR I = 17 TO 5 STEP -1
```

This program prints a rocket countdown:

```
10 FOR I = 10 TO 1 STEP -1
20     PRINT I
30 NEXT
40 PRINT "BLAST OFF!"
```

The computer will print:

```
 10
 9
 8
 7
 6
 5
 4
 3
 2
 1
BLAST OFF!
```

This statement is tricky:

```
FOR I = 5 TO 16 STEP 3
```

It says to start I at 5, and keep adding 3 until I gets past 16. So I will be 5, then 8, then 11, then 14. I won't be 17, since 17 is past 16. The first value of I is 5; the last value is 14.

In the statement FOR I = 5 TO 16 STEP 3, the *first value* or *initial value* of I is 5, the *limit value* is 16, and the *step size* or *increment* is 3. The I is called the *counter* or *index* or *loop-control variable*. Although the limit value is 16, the *last value* or *terminal value* is 14.

Programmers usually say "FOR I", instead of "FOR X", because the letter I reminds them of the word *index*.

PRETTY OUTPUT

ZONES

If your computer's fancy, the leftmost part of the screen is called the first *zone*; to the right of that zone lies the second zone; to the right of the second zone lies the third zone; then comes the fourth zone; and the rightmost part of the screen is called the fifth zone. So altogether, there are 5 zones. Each zone is 16 characters wide; the entire screen is 80 characters wide.

(If your computer's *not* so fancy, it might have fewer zones, or the zones might be narrower; and the screen might be less than 80 characters wide. To find out about *your* computer's peculiarities, read the appendix.)

A comma makes the computer jump to a new zone; here's an example:

```
PRINT "SIN","KING"
```

The computer will print SIN and KING on the same line; but because of the comma before KING, the computer will print KING in the second zone, like this:

```
SIN             KING
```

first zone *second zone* *third zone* *fourth zone* *fifth zone*

Of course, you can turn that example into a program, by putting a number in front of the line, like this:

```
10 PRINT "SIN","KING"
```

When you type RUN, the computer will print:

```
SIN             KING
```

This program does the same thing:

```
10 PRINT "SIN",
20 PRINT "KING"
```

Line 10 makes the computer print SIN and then jump to the next zone. Line 20 makes the computer print KING. The computer will print:

```
SIN             KING
```

This example's silly:

```
PRINT "LOVE","CRIES","OUT"
```

The computer will print LOVE in the first zone, CRIES in the second zone, and OUT in the third zone, like this:

```
LOVE            CRIES           OUT
```

This example's even sillier:

```
PRINT "LOVE","CRIES","OUT","TO","ME","AT","NIGHT"
```

The computer will print LOVE in the first zone, CRIES in the second, OUT in the third, TO in the fourth, ME in the fifth, and the remaining words below, like this:

```
LOVE            CRIES           OUT             TO              ME
AT              NIGHT
```

This program makes the computer greet you:

```
10 PRINT "HI",
20 GO TO 10
```

The computer will print HI many times; each time will be in a new zone, like this:

```
HI            HI            HI            HI            HI
HI            HI            HI            HI            HI
HI            HI            HI            HI            HI
```
etc.

This program prints a list of words and their opposites:

```
10 PRINT "GOOD","BAD"
20 PRINT "BLACK","WHITE"
30 PRINT "GRANDPARENT","GRANDCHILD"
40 PRINT "HE","SHE"
```

Line 10 makes the computer print GOOD, then jump to the next zone, then print BAD. Altogether, the computer will print:

```
GOOD          BAD
BLACK         WHITE
GRANDPARENT   GRANDCHILD
HE            SHE
```

The first zone contains a column of words; the second zone contains the opposites. Altogether, the computer's printing looks like a table. So whenever you want to make a table, use zones, by putting commas in your program.

Let's make the computer print this table:

```
NUMBER        SQUARE
 3             9
 4             16
 5             25
 6             36
 7             49
 8             64
 9             81
 10            100
```

Here's the program:

```
10 PRINT "NUMBER","SQUARE"
20 FOR I = 3 TO 10
30     PRINT I,I*I
40 NEXT
```

Line 10 prints the word NUMBER at the top of the first column, and the word SQUARE at the top of the second. Line 20 says I goes from 3 to 10; to begin, I is 3. Line 30 makes the computer print:

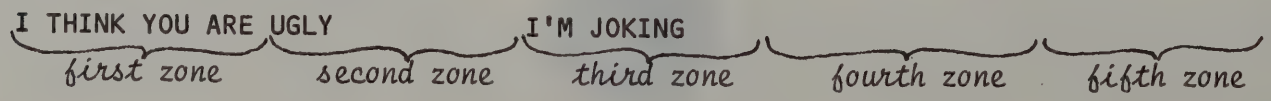
```
 3             9
```

Line 40 makes the computer do the same thing for the next I, and for the next I, and for the next; so the computer prints the whole table.

This example tells a bad joke:

```
PRINT "I THINK YOU ARE UGLY","I'M JOKING"
```

The computer will print I THINK YOU ARE UGLY, then jump to a new zone, then print I'M JOKING, like this:



You can make the computer skip over a zone:

```
PRINT "JOE"," ","LOVES SUE"
```

The computer will print JOE in the first zone, a blank space in the second zone, and LOVES SUE in the third zone, like this:

```
JOE                     LOVES SUE
```

first zone *second zone* *third zone* *fourth zone* *fifth zone*

You can type that example even more briefly, like this:

```
PRINT "JOE",,"LOVES SUE"
```

When you combine commas with semicolons, you can get weird results:

```
PRINT "EAT","ME";"AT";"BALLS","NO";"W"
```

That line contains commas and semicolons. A comma makes the computer jump to a new zone, but a semicolon does *not* make the computer jump. The computer will print EAT, then jump to a new zone, then print ME and AT and BALLS, then jump to a new zone, then print NO and W. Altogether, the computer will print:

```
EAT             MEATBALLS       NOW
```

TAB

When the computer puts a line of information on your screen, the leftmost character in the line is said to be at *position 1*; the second character in the line is said to be at *position 2*. This command makes the computer skip to position 6 and then print "HOT":

```
PRINT TAB(6)"HOT"
```

The computer will print:

```
     HOT
12345678
```

Here's a fancier example:

```
PRINT TAB(6)"HOT";TAB(13)"BUNS"
```

The computer will skip to the 6th position, then print "HOT", then skip to the 13th position, then print "BUNS":

```
     HOT    BUNS
123456789101112131415 16
```

This program prints a diagonal line:

```
10 FOR I = 1 TO 12
20     PRINT TAB(I)"*"
30 NEXT
```

Line 10 says to do the loop twelve times; so the computer does line 20 repeatedly. The first time the computer does line 20, the I is 1, so the computer prints an asterisk at position 1:

```
*
```

The next time, the I is 2, so the computer skips to position 2 and prints an asterisk:

```
 *
```

The next time, the I is 3, so the computer skips to position 3 and prints an asterisk:

```
  *
```

Altogether, the program makes the computer print this picture:

```
*
 *
  *
   *
    *
     *
      *
       *
        *
         *
          *
           *
```

CALENDAR

Let's make the computer print a calendar for the whole year. To begin, we must tell the computer how many days are in each month:

Month's name	Month's length
January	31
February	28 or 29
March	31
April	30
May	31
June	30
July	31
August	31
September	30
October	31
November	30
December	31

That list becomes our data:

```
10 DATA JANUARY,31,FEBRUARY,28,MARCH,31,APRIL,30
20 DATA MAY,31,JUNE,30,JULY,31,AUGUST,31
30 DATA SEPTEMBER,30,OCTOBER,31,NOVEMBER,30,DECEMBER,31
```

(For a leap year, change line 10.)

To make the computer look at that data, tell the computer to READ:

```
40 READ N$,L
```

That makes the computer read a month's Name and Length.

To read *all* the data, the computer should do line 40 twelve times. So put line 40 inside a loop, like this:

Here are the months and their lengths.

```
10 DATA JANUARY,31,FEBRUARY,28,MARCH,31,APRIL,30
20 DATA MAY,31,JUNE,30,JULY,31,AUGUST,31
30 DATA SEPTEMBER,30,OCTOBER,31,NOVEMBER,30,DECEMBER,31
```

For each month,

```
39 FOR M = 1 TO 12
```

read its Name & Length,

```
40     READ N$,L
```

print its Name,

```
50     PRINT N$
```

print all its days,

```
60     FOR D = 1 TO L
70         PRINT D;
80     NEXT
```

and press the ENTER key twice at end of month.

```
90     PRINT
91     PRINT
100 NEXT
```

The computer will print a calendar beginning like this:

```
JANUARY
 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
 21  22  23  24  25  26  27  28  29  30  31

FEBRUARY
 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
 21  22  23  24  25  26  27  28

MARCH
 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
 21  22  23  24  25  26  27  28  29  30  31
```

Although the computer's printed the right numbers for each month, the numbers are in the wrong places. Let's make the computer print at most 7 numbers in each row, so each is a week.

To print the numbers in the right places, use TAB:

```
70         PRINT TAB(T)D;
```

Line 70 will make the computer print each day in the right position . . . if we define T correctly. But how should we define T?

For Sunday, let's make T be 2 (so that Sunday begins at position 2); for Monday, let's make T be 6 (so that Monday begins at position 6); for Tuesday, let's make T be 10; for Wednesday, 14; Thursday, 18; Friday, 22; and Saturday, 26. So whenever a day's been printed, T should normally increase by 4, for the next day:

```
71         T=T+4
```

Saturday's the last day of the week; after Saturday, we must begin a new week. So if T has passed Saturday (which is 26), we want T to become 2 (for Sunday); and if there are more days left in the month, we want the computer to press the ENTER key (to start a new week):

```
72         IF T>26 THEN T=2: IF D<L THEN PRINT
```

Which year would you like a calendar for: 1985? 1986? 1987? This program makes a pretty calendar for 1986:

```
→→1 PRINT "CALENDAR FOR 1986"
→→2 PRINT
→→3 T=14
  10 DATA JANUARY,31,FEBRUARY,28,MARCH,31,APRIL,30
  20 DATA MAY,31,JUNE,30,JULY,31,AUGUST,31
  30 DATA SEPTEMBER,30,OCTOBER,31,NOVEMBER,30,DECEMBER,31
  39 FOR M = 1 TO 12
  40     READ N$,L
  50     PRINT N$
→→51     PRINT "  SUN MON TUE WED THU FRI SAT"
  60     FOR D = 1 TO L
  70         PRINT TAB(T)D;
  71         T=T+4
  72         IF T>26 THEN T=2: IF D<L THEN PRINT
  80     NEXT
  90     PRINT
  91     PRINT
  100 NEXT
```

Line 1 prints the heading. Line 2 puts a blank line underneath the heading. Since 1986 begins on a Wednesday, line 3 tells the computer to start T at 14 (which is the position for Wednesday). Line 51 prints the heading for each month; when you type that line, put 2 blank spaces before SUN. The computer will print a calendar beginning like this:

```
CALENDAR FOR 1986

JANUARY
  SUN MON TUE WED THU FRI SAT
                1   2   3   4
  5   6   7   8   9   10  11
  12  13  14  15  16  17  18
  19  20  21  22  23  24  25
  26  27  28  29  30  31

FEBRUARY
  SUN MON TUE WED THU FRI SAT
                        1
  2   3   4   5   6   7   8
  9   10  11  12  13  14  15
  16  17  18  19  20  21  22
  23  24  25  26  27  28

MARCH
  SUN MON TUE WED THU FRI SAT
                        1
  2   3   4   5   6   7   8
  9   10  11  12  13  14  15
  16  17  18  19  20  21  22
  23  24  25  26  27  28  29
  30  31
```

If you want a different year, change lines 1 and 3; and if you want a leap year, change line 10. If you want the calendar to be taller, insert extra blank lines; to do that, replace "PRINT" by "PRINT: PRINT: PRINT" in lines 2, 72, 90, and 91:

```
2 PRINT: PRINT: PRINT
72          IF T>26 THEN T=2: IF D<L THEN PRINT: PRINT: PRINT
90      PRINT: PRINT: PRINT
91      PRINT: PRINT: PRINT
```

If you want the calendar to look wider (or narrower), change the positions for Sunday, Monday, Tuesday, etc., by changing the T numbers (in lines 3, 51, 71, and 72).

LOCATE

The computer makes the screen show several lines of information. On the screen, the top line is called *line 1*; underneath it is *line 2*; then comes *line 3*; etc.

Each line consists of many characters. The leftmost character is at *position 1*; the next character is at *position 2*; etc.

On the screen, the computer will print wherever you wish. For example, to make the computer print the word DROWN, so that the word DROWN begins at line 3's 7th position, type this:

```
LOCATE 3,7: PRINT "DROWN"
```

The computer will print the word's first letter (D) at line 3's 7th position. The computer will print the rest of the word afterwards.

You'll see the first letter (D) at line 3's 7th position, the next letter (R) at the next position (line 3's 8th position), the next letter (O) at the next position (line 3's 9th position), etc.

Most computers understand the word LOCATE, but *your* computer might be different. To find out about *your* computer, check the appendix.

PIXELS

The image on the computer's screen is called the *picture*. If you stare at the picture closely, you'll see the picture's composed of thousands of tiny rectangles. Each tiny rectangle is called a *picture's element*, or *pic's el*, or *pixel*, or *pel*.

The tiny rectangle in the screen's upper-left corner is called pixel (0,0); just to the right of it is pixel (1,0); then comes pixel (2,0); etc. Underneath pixel (0,0) is pixel (0,1). Here are the positions of the pixels:

pixel (0,0)	*pixel (1,0)*	*pixel (2,0)*	*pixel (3,0)*	*pixel (4,0)*	*etc.*
pixel (0,1)	*pixel (1,1)*	*pixel (2,1)*	*pixel (3,1)*	*pixel (4,1)*	*etc.*
pixel (0,2)	*pixel (1,2)*	*pixel (2,2)*	*pixel (3,2)*	*pixel (4,2)*	*etc.*
pixel (0,3)	*pixel (1,3)*	*pixel (2,3)*	*pixel (3,3)*	*pixel (4,3)*	*etc.*

Each pixel's name consists of two numbers in parentheses. The first number's called the *X coordinate*; the second number's called the *Y coordinate*. For example, if you're talking about pixel (4,3), its X coordinate is 4, and its Y coordinate is 3.

The X coordinate tells how far to the right the pixel is. The Y coordinate tells how far down. So pixel (4,3) is the pixel that's 4 to the right and 3 down.

On the computer, the Y coordinate measures how far *down*, not up! If you've have the misfortune of reading old-fashioned math books in which the Y coordinate measured how far up, you'll have to reverse your thinking!

How many pixels are on the screen? The answer depends on which computer you have. If your computer's typical, the X coordinate goes from 0 to 319, and the Y coordinate goes from 0 to 199. So on a typical computer, the pixel at the screen's lower right-hand corner is pixel (319,199); and the total number of pixels on the screen is "320 times 200", which is 64 thousand. But *your* computer might not be "typical"; check the appendix.

Fundamental shapes. The computer can draw three fundamental shapes: dots, lines, and circles.

To make the computer draw a dot at pixel (100,100), say:

```
PLOT (100,100)
```

(Although most computers understand the word PLOT, *some* computers require you to use a different word instead, such as "DRAW" or "PSET". To find out which word *your* computer understands, check the appendix.)

To make the computer draw a line from pixel (0,0) to pixel (100,100), say:

```
LINE (0,0)-(100,100)
```

To make the computer draw a line from pixel (0,0) to pixel (100,100), and then draw a line from that pixel (100,100) to pixel (70,120), say:

```
LINE (0,0)-(100,100)
LINE -(70,120)
```

To make the computer draw a circle whose center is pixel (100,100) and whose radius is 50, say:

```
CIRCLE (100,100),50
```

The computer draws each shape in white, on a black background.

Most computers understand those commands, but *your* computer might be different; check the appendix.

Painting. After you've drawn an outline of a shape (by using dots, lines, and circles), you can fill in the middle of the shape, by telling the computer to PAINT the shape.

Here's how to PAINT a shape that you've drawn (such as a circle or a house). Find a pixel that's in the middle of the shape and that's still black; then tell the computer to PAINT, starting at that pixel. For example, if pixel (100,101) is inside the shape and is still black, say:

```
PAINT (100,101)
```

Colors. You can use these colors:

0. *black*
1. *blue*
2. *green*
3. *cyan (greenish blue)*
4. *red*
5. *magenta (purplish red)*
6. *brown*
7. *cream (yellowish white)*
8. *light black (gray)*
9. *light blue*
10. *light green*
11. *light cyan (aqua)*
12. *light red (pink)*
13. *light magenta*
14. *light brown (yellow)*
15. *light cream (pure white)*

Normally, the PLOT, LINE, CIRCLE, and PAINT commands draw in yellowish white (cream). If you prefer a different color, put a comma and the color's number at the end of the command. For example, if you want to draw a line from (0,0) to (100,0) in green (whose color number is 2), type this:

```
LINE (0,0)-(100,0),2
```

When you give a PAINT command, you must make its color the same as the color of the outline you're filling in.

Because American TV's are crummy, you'll get the correct colors only if you draw horizontal lines. If you draw vertical lines or diagonals or circles or dots, the colors will be slightly off.

The horizontal lines will be perfect colors. If they don't look perfect to you adjust your TV's COLOR and TINT dials.

Boxes. If you type—

```
LINE (0,0)-(100,100),2
```

the computer draws a line from pixel (0,0) to (100,100) using color 2.

If you put the letter B at the end of the line command, like this—

```
LINE (0,0)-(100,100),2,B
```

the computer will draw a box instead of a line. One corner of the box will be at pixel (0,0); the opposite corner will be at (100,100); and the box will be drawn using color 2.

If you put BF at the end of the line command, like this—

```
LINE (0,0)-(100,100),2,BF
```

the computer will draw a box and also fill it in, by painting its interior.

To produce sounds, you can say BEEP, SOUND, or PLAY. BEEP appeals to business executives; SOUND appeals to doctors and engineers; and PLAY appeals to musicians. I'll explain how the typical computer handles the words BEEP, SOUND, and PLAY, but check the appendix to find out whether *your* computer handles them differently.

BEEP. If you type—

```
BEEP
```

the typical computer will beep. Specifically, it will play a note whose frequency ("pitch") is 800 hertz, and it will play the note for a quarter of a second.

You can say BEEP in the middle of your program. For example, you can tell the computer to BEEP if a person enters wrong data.

This program makes the computer acts as a priest and perform a marriage ceremony:

```
10 INPUT "DO YOU TAKE THIS WOMAN TO BE YOUR LAWFUL WEDDED WIFE";A$
20 IF A$<>"I DO" THEN BEEP: PRINT "TRY AGAIN!": GO TO 10
30 INPUT "DO YOU TAKE THIS MAN TO BE YOUR LAWFUL WEDDED HUSBAND";A$
40 IF A$<>"I DO" THEN BEEP: PRINT "TRY AGAIN!": GO TO 30
50 PRINT "I NOW PRONOUNCE YOU HUSBAND AND WIFE."
```

Line 10 makes the computer ask the groom, "DO YOU TAKE THIS WOMAN TO BE YOUR LAWFUL WEDDED WIFE?" If the groom doesn't say "I DO", line 20 makes the computer beep, say "TRY AGAIN!", and repeat the question. Lines 30 and 40 do the same thing to the bride. Line 50 congratulates the couple for having answered correctly.

The SOUND command. If you type—

```
SOUND 440,18.2
```

the typical computer will produce a sound. In that command, the 440 is the frequency ("pitch"), measured in hertz (cycles per second); so the sound will be a musical note whose pitch is 440 hertz. (That note happens to be "the A above middle C".)

If you replace the 440 by a higher number, the sound will have a higher pitch.

When you were a baby, you could probably hear up to 20000; as you get older, your hearing gets worse, and you can't hear such high notes. Today, the highest sound you can hear is probably somewhere around 14000.

To find out, give yourself a hearing test, by running this program:

```
10 INPUT "WHAT PITCH WOULD YOU LIKE ME TO PLAY";P
20 SOUND P,18.2
30 GO TO 10
```

When you run that program, begin by inputting a low pitch (such as 200). Then input a higher number, then an even higher number, until you finally pick a number so high you can't hear it. (When trying that test, put your ear close to the computer's speaker, which is in the computer's front left corner.) When you've picked a number too high for you to hear, try a slightly lower number. Keep trying different numbers, until you find the highest number you can hear.

I'm 38 years old, and the highest pitch *I* can hear is about 14000. How about *you*? Is your hearing better than mine?

Have a contest with your friends: find out which of your friends can hear best.

If you run that program every year, you'll see that your hearing gets gradually worse.

In those examples, the 18.2 makes the computer produce the sound for 1 second. If you want the sound to last longer—so that it lasts 2 seconds—replace the 18.2 by 18.2*2. For 10 seconds, say 18.2*10. (That's because the computer's metronome beats 18.2 times per second.)

PLAY. If you type—

```
PLAY "CDG#B-A"
```

the typical computer will play the note C, then D, then G sharp, then B flat, then A.

The computer can play in seven octaves, numbered from 0 to 6. Octave 0 consists of very bass notes; octave 6 consists of very high-pitched notes. In each octave, the lowest note is a C: the notes in an octave are C, C#, D, D#, E, F, F#, G, G#, A, A#, and B.

"Middle C" is at the beginning of octave 2. Normally, the computer plays in octave 4. To make the computer switch to octave 3, type *the letter "O"* followed by a 3, like this:

```
PLAY "O3"
```

After giving that command, anything else you PLAY will be in octave 3, until you change octaves again.

Besides playing with pitches, you can also play with rhythms ("lengths" of the notes). Normally each note is a "quarter note". To make the computer switch to eighth notes (which are faster), type this:

```
PLAY "L8"
```

Besides using L8 for eighth notes, you can use L16 for sixteenth notes (which are even faster), L32 for thirty-second notes (which are super-fast), and L64 for sixty-fourth notes (which are super-super-fast). For long notes, you can use L2 (which gives a half note) or L1 (which gives a whole note).

You can use any length from L1 to L64. You can even use in-between lengths, such as L7 or L23 (though such rhythms are hard to stamp your foot to).

If you put a period after the length, the computer will multiply the length by 1½. For example, if you say "L8." instead of just "L8", the computer will play the notes 1½ times as long as 8th notes. Musicians call them "dotted eighth notes."

To make the computer pause ("rest") for an eighth note, put a P8 into the music string.

Normally, the computer plays 120 quarter notes per minute; but you can change that tempo. To switch to 150 quarter notes per minute, say:

```
PLAY "T150"
```

You can switch to any tempo from 32 to 255. The 32 is very slow; 255 is very fast. In musical terms, 40=larghissimo, 50=largo, 63=larghetto, 65=grave, 68=lento, 71=adagio, 76=andantino, 92=andante, 114=moderato, 120=allegretto, 144=allegro, 168=vivace, 188=presto, and 208=prestissimo.

You can combine all those musical commands into a single PLAY statement. For example, to set the tempo to 150, the octave to 3, the length to 8 (which means an eighth note), and then play C and D, and then change the length to 4 and play E, type this:

```
PLAY "T150O3L8CDL4E"
```

ELEMENTS

CONSTANTS

A *numeric constant* is a simple number, such as:

```
0   1   2   8   43.7   -524.6   .003   1.3E5
                                      ↑
                          which means, "take 1.3, and move the
                          decimal point 5 places to the right"
```

A numeric constant does not contain any arithmetic. For example, since 7+1 contains arithmetic (+), it's *not* a numeric constant. 8 is a numeric constant, even though 7+1 isn't.

A *string constant* is a simple string, in quotation marks:

```
"I LOVE YOU"   "76 TROMBONES"   "GO AWAY!!!"   "XYPW EXR///746"
```

A *constant* is a numeric constant or a string constant:

```
0   8   -524.6   1.3E5   "I LOVE YOU"   "XYPW EXR///746"
```

KINDS OF VARIABLES

A *string variable* is a variable that stands for a string:

```
A$   B$   Y$   Z$
```

A *numeric variable* is a variable that stands for a number:

```
A   B   Y   Z
```

A *variable* is a string variable or a numeric variable:

```
A$   Z$   A   Z
```

EXPRESSIONS

A *numeric expression* is a numeric constant (such as 8) or a numeric variable (such as A) or a combination of them, such as 8+Z, or 8*A, or Z*A, or 8*2, or 7+1, or even Z*A-(7+Z)/8+1.3E5*(-524.6+B).

A *string expression* is a string constant (such as "I LOVE YOU") or a string variable (such as A$) or a combination.

An *expression* is a numeric expression or a string expression.

STATEMENTS

In a DATA statement, you must have constants.

```
Right:  10 DATA 8, 1.3E5       (8 and 1.3E5 are constants.)
Wrong:  10 DATA 7+1, 1.3E5     (7+1 is not a constant.)
```

In the DATA statement, if the constant is a string, you can omit the quotation marks (unless the string contains a comma or a colon).

```
Right:       10 DATA "JOE","MARY"
Also right:  10 DATA JOE,MARY
```

The INPUT statement's prompt must be a string constant.

```
Right:  10 INPUT "WHAT IS YOUR NAME";N$     ("WHAT IS YOUR NAME" is a constant.)
Wrong:  10 INPUT Q$;N$                      (Q$ is not a constant.)
```

In a GO TO statement, the line number must be a numeric constant.

```
Right:  50 GO TO 100      (100 is a numeric constant.)
Wrong:  50 GO TO N        (N is not a numeric constant.)
```

Here are the forms of the most popular BASIC statements:

General form	Example
PRINT list of expressions and commas and semicolons and tabs	PRINT TAB(6)"HOT",2+2;"U"
GO TO numeric constant	GO TO 10
END	END
STOP	STOP
variable = expression	X = 47+2
INPUT string constant ; variable	INPUT "WHAT IS YOUR NAME";N$
IF condition THEN list of statements	IF A>=18 THEN PRINT "YOU": PRINT "VOTE"
DATA list of constants	DATA JOE,273,219,MARY,412,371
READ list of variables	READ N$,B,A
RESTORE	RESTORE
FOR numeric variable = numeric expression TO numeric expression STEP numeric expression	FOR I = 59+1 TO 100+N STEP 2+3
NEXT numeric variable	NEXT I

A VARIABLE IS A BOX

The computer's random-access memory (RAM) consists of electronic boxes. This program puts a number into a box:

```
10 X=47
20 PRINT X+2
```

Line 10 puts 47 into box X, like this:

Box X [47]

Line 20 says to print what's in box X, plus 2. So the computer will print 49.

You can change what's in a box:

```
10 F=4
20 F=9
30 PRINT F*2
```

Line 10 puts 4 into box F:

Box F [4]

Line 20 puts 9 into box F; the 9 replaces the 4:

Box F [9]

Line 30 prints 18.

Saying "A=B" has a different effect than "B=A". Compare these programs:

```
10 A=1              10 A=1
20 B=7              20 B=7
30 A=B              30 B=A
40 PRINT A+B        40 PRINT A+B
```

In both programs, lines 10 and 20 do this:

Box A [1]
Box B [7]

In the left program, line 30 makes the number in box A become 7, so both boxes contain 7; line 40 prints 14. In the right program, line 30 makes the number in box B become 1, so both boxes contain 1; line 40 prints 2.

Try this one-line program:

```
10 PRINT R
```

That line says to print a *number*—the number in box R. Since box R is empty, the computer prints this number:

```
 0
```

LOOP TECHNIQUES

INCREMENTING

Here's a strange program:

```
10 A=5
20 A=3+A
30 PRINT A
```

Line 20 means: the new A is 3 plus the old A. So the new A is 3+5, which is 8. Line 30 prints:

```
 8
```

Let's look at that program more closely. Line 10 puts 5 into box A:

Box A [5]

When the computer sees line 20, it examines the right side of the equation, and sees the 3+A. Since A is 5, the 3+A is 3+5, which is 8. So line 20 says: A=8. The computer puts 8 into box A:

Box A [8]

Line 30 prints 8.

Here's another weirdo:

```
10 B=6
20 B=B+1
30 PRINT B*2
```

Line 20 says the new B is "the old B plus 1". So the new B is 6+1, which is 7. Line 30 prints:

```
 14
```

In that program, line 10 says B is 6; but line 20 increases B, by adding 1 to B; so B becomes 7. Programmers say that B has been *increased* or *incremented*. In line 20, the "1" is called the *increase* or the *increment*.

DECREMENTING

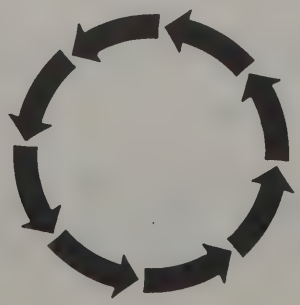
The opposite of *increment* is *decrement*:

```
10 J=500
20 J=J-1
30 PRINT J
```

Line 10 says J starts at 500. But line 20 says the new J is "the old J minus 1", so the new J is 500-1, which is 499. Line 30 prints:

```
 499
```

In that program, J was *decreased* (or *decremented*). In line 20, the "1" is called the *decrease* (or *decrement*).



Suppose you want the computer to count, starting at 3, like this:

```
3
4
5
6
7
8
etc.
```

This program does it, by a special technique:

```
10 C=3
20 PRINT C
30 C=C+1
40 GO TO 20
```

In that program, C is called the *counter*, because it helps the computer count.

Line 10 says C starts at 3. Line 20 makes the computer print C, so the computer prints:

```
3
```

Line 30 increases C, by adding 1 to it, so C becomes 4. Line 40 sends the computer back to line 20, which prints the new value of C:

```
4
```

Then the computer comes to line 30 again, which increases C again, so C becomes 5. Line 40 sends the computer back to line 20 again, which prints:

```
5
```

The program is an infinite loop: the computer will print 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, and so on, forever, unless you abort it.

General procedure. Here's the general procedure for making the computer count:

1. *Start C at some value (such as 3).*
2. *Use C. (For example, tell the computer to PRINT C.)*
3. *Increase C, by saying C=C+1.*
4. *GO back TO step 2.*

Variations. To read the printing more easily, put a semicolon at the end of the PRINT statement:

```
  10 C=3
→→20 PRINT C;
  30 C=C+1
  40 GO TO 20
```

The semicolon makes the computer print horizontally:

```
3  4  5  6  7  8  etc.
```

This program makes the computer count, starting at 1:

```
→→10 C=1
  20 PRINT C;
  30 C=C+1
  40 GO TO 20
```

The computer will print 1, 2, 3, 4, etc.

This program makes the computer count, starting at 0:

```
→→10 C=0
  20 PRINT C;
  30 C=C+1
  40 GO TO 20
```

The computer will print 0, 1, 2, 3, 4, etc.

Tables. Let's make the computer print a table, showing each number and its square, like this:

```
NUMBER          SQUARE
 0               0
 1               1
 2               4
 3               9
 4               16
 5               25
 6               36
 7               49
 8               64
 etc.
```

This program does it:

```
→→5 PRINT "NUMBER","SQUARE"
  10 C=0
→→20 PRINT C,C*C
  30 C=C+1
  40 GO TO 20
```

Line 5 prints the headings. Lines 10-40 make the computer count (0, 1, 2, 3, etc.); but instead of printing just C, line 20 also prints C*C.

Let's make the computer print a table that shows the squares of decimals, like this:

```
NUMBER          SQUARE
 0               0
 .1              .01
 .2              .04
 .3              .09
 .4              .16
 .5              .25
 .6              .36
 .7              .49
 .8              .64
 .9              .81
 1               1
 1.1             1.21
 1.2             1.44
 1.3             1.69
 1.4             1.96
 1.5             2.25
 etc.
```

To make the computer do that, just change line 30, so that the computer increases by .1 instead of by 1:

```
  5 PRINT "NUMBER","SQUARE"
  10 C=0
  20 PRINT C,C*C
→→30 C=C+.1
  40 GO TO 20
```

Let's make the computer give this quiz:

What's the capital of Nevada?
What's the chemical symbol for iron?
What word means 'brother or sister'?
What was Beethoven's first name?
How many cups are in a quart?

To make the computer score the quiz, we must tell it the correct answers, which are:

Carson City
Fe
sibling
Ludwig
4

So the program contains this data:

```
10 DATA WHAT'S THE CAPITAL OF NEVADA,CARSON CITY
20 DATA WHAT'S THE CHEMICAL SYMBOL FOR IRON,FE
30 DATA WHAT WORD MEANS 'BROTHER OR SISTER',SIBLING
40 DATA WHAT WAS BEETHOVEN'S FIRST NAME,LUDWIG
50 DATA HOW MANY CUPS ARE IN A QUART,4
```

Tell the computer to READ the data:

```
100 READ Q$,A$
```

That line reads a pair of data: it reads a question (Q$) and the correct answer (A$).

Make the computer ask the question, and wait for the human's response:

```
110 PRINT Q$;
120 INPUT "??";H$
```

Line 110 prints the question. Line 120 prints question marks after the question, and waits for the human to respond; the human's response is called H$.

Finally, evaluate the human's response. If the human's response (H$) is the correct answer (A$), make the computer say "CORRECT" and GO TO the next question:

```
130 IF H$=A$ THEN PRINT "CORRECT": GO TO 100
```

But if the human's response is wrong, make the computer say "NO" and reveal the correct answer:

```
140 PRINT "NO, THE ANSWER IS: ";A$: GO TO 100
```

Here's a sample run:

```
RUN
WHAT'S THE CAPITAL OF NEVADA??? LAS VEGAS
NO, THE ANSWER IS: CARSON CITY
WHAT'S THE CHEMICAL SYMBOL FOR IRON??? FE
CORRECT
WHAT WORD MEANS 'BROTHER OR SISTER'??? I GIVE UP
NO, THE ANSWER IS: SIBLING
WHAT WAS BEETHOVEN'S FIRST NAME??? LUDVIG
NO, THE ANSWER IS: LUDWIG
HOW MANY CUPS ARE IN A QUART??? 4
CORRECT
OUT OF DATA
```

To give a quiz about different topics, change the data in lines 10-50.

Instead of making the computer say OUT OF DATA, let's make it say:

```
I HOPE YOU ENJOYED THE QUIZ
```

To do that, write an end mark and an end routine:

```
60 DATA END,END
100 READ Q$,A$: IF Q$="END" THEN PRINT "I HOPE YOU ENJOYED THE QUIZ": END
```

Counting the correct answers. Let's make the computer count how many questions the human answered correctly. To do that, we need a counter. As usual, we'll call it C:

```
→→5 C=0
  10 DATA WHAT'S THE CAPITAL OF NEVADA,CARSON CITY
  20 DATA WHAT'S THE CHEMICAL SYMBOL FOR IRON,FE
  30 DATA WHAT WORD MEANS 'BROTHER OR SISTER,SIBLING
  40 DATA WHAT WAS BEETHOVEN'S FIRST NAME,LUDWIG
  50 DATA HOW MANY CUPS ARE IN A QUART,4
  60 DATA END,END
→→100 READ Q$,A$: IF Q$="END" THEN PRINT "YOU ANSWERED";C;"OF THE QUESTIONS CORR
  ECTLY": PRINT "I HOPE YOU ENJOYED THE QUIZ": END
  110 PRINT Q$;
  120 INPUT "??";H$
→→130 IF H$=A$ THEN PRINT "CORRECT": C=C+1: GO TO 100
  140 PRINT "NO, THE ANSWER IS: ";A$: GO TO 100
```

At the beginning of the program, the human hasn't answered any questions correctly yet, so line 5 begins the counter at 0. Each time the human answers a question correctly, line 130 increases the counter. When the program ends, line 100 prints the counter, by printing a message such as:

```
YOU ANSWERED 2 OF THE QUESTIONS CORRECTLY
```

Printing the score. It would be nicer to print—

```
YOU ANSWERED 2 OF THE 5 QUESTIONS CORRECTLY
YOUR SCORE IS 40 %
```

or, if the quiz were changed to include 8 questions:

```
YOU ANSWERED 2 OF THE 8 QUESTIONS CORRECTLY
YOUR SCORE IS 25 %
```

To make the computer print the 5 or the 8, make the computer count how many questions were asked. So we need another counter. Since we already used C to count the number of correct answers, let's use Q to count the number of questions asked. Like C, Q must start at 0; and we must increase Q, by adding 1 each time another question is asked:

```
→→5 C=0: Q=0
  10 DATA WHAT'S THE CAPITAL OF NEVADA,CARSON CITY
  20 DATA WHAT'S THE CHEMICAL SYMBOL FOR IRON,FE
  30 DATA WHAT WORD MEANS 'BROTHER OR SISTER',SIBLING
  40 DATA WHAT WAS BEETHOVEN'S FIRST NAME,LUDWIG
  50 DATA HOW MANY CUPS ARE IN A QUART,4
  60 DATA END,END
→→100 READ Q$,A$: IF Q$="END" THEN PRINT "YOU ANSWERED";C;"OF THE";Q;"QUESTIONS
  CORRECTLY": PRINT "YOUR SCORE IS";C/Q*100;"%": PRINT "I HOPE YOU ENJOYED THE Q
  UIZ": END
  110 PRINT Q$;
→→115 Q=Q+1
  120 INPUT "??";H$
  130 IF H$=A$ THEN PRINT "CORRECT": C=C+1: GO TO 100
  140 PRINT "NO, THE ANSWER IS: ";A$: GO TO 100
```

SUMMING

Let's make the computer imitate an adding machine, so a run looks like this:

```
RUN
NOW THE SUM IS 0
WHAT NUMBER DO YOU WANT TO ADD TO THE SUM? 5
NOW THE SUM IS 5
WHAT NUMBER DO YOU WANT TO ADD TO THE SUM? 3
NOW THE SUM IS 8
WHAT NUMBER DO YOU WANT TO ADD TO THE SUM? 6.1
NOW THE SUM IS 14.1
WHAT NUMBER DO YOU WANT TO ADD TO THE SUM -10
NOW THE SUM IS 4.1
etc.
```

Here's the program:

```
10 S=0
20 PRINT "NOW THE SUM IS";S
30 INPUT "WHAT NUMBER DO YOU WANT TO ADD TO THE SUM";X
40 S=S+X
50 GO TO 20
```

Line 10 starts the sum at 0. Line 20 prints the sum. Line 30 asks the human what number to add to the sum; the human's number is called X. Line 40 adds X to the sum, so the sum changes. Line 50 makes the computer go to line 20, which prints the new sum. Lines 20-50 form an infinite loop, which you must abort.

Here's the general procedure for making the computer find a sum:

1. *Start S at 0.*
2. *Use S. (For example, tell the computer to PRINT S.)*
3. *Find out what number to add to S. (For example, let the human input an X.)*
4. *Increase S, by saying S = S + the number to be added.*
5. *GO back TO step 2.*

CHECKING ACCOUNT

If your bank's nasty, it charges you 10¢ for processing each good check that you write, and a $5 penalty for each check that bounces; and it doesn't pay any interest on the money you've deposited. This program makes the computer imitate such a bank:

Start the sum at 0→→→
```
10 S=0
```
Chat with the human→→
```
20 PRINT "YOUR CHECKING ACCOUNT CONTAINS";S
30 INPUT "DEPOSIT OR WITHDRAW";A$
40 IF A$="DEPOSIT" THEN GO TO 100
50 IF A$="WITHDRAW" THEN GO TO 200
60 PRINT "PLEASE SAY DEPOSIT OR WITHDRAW": GO TO 30
```
Deposit some money→→→
```
100 INPUT "HOW MUCH DO YOU WANT TO DEPOSIT";D
110 S=S+D
120 GO TO 20
```
Withdraw some money→→
```
200 INPUT "HOW MUCH DO YOU WANT TO WITHDRAW";W
210 W=W+.10
220 IF W<=S THEN PRINT "OKAY": S=S-W: GO TO 20
230 PRINT "THAT CHECK BOUNCED": S=S-5: GO TO 20
```

In that program, the total amount of money in the checking account is called the sum. Line 10 starts that sum at 0. Line 20 prints the sum. Line 30 asks whether the human wants to deposit or withdraw.

If the human says DEPOSIT, the computer goes from line 40 to line 100 (which asks how much to deposit), then to line 110 (which adds the deposit to the sum in the account), then back to line 20 (for the next transaction).

But if the human says WITHDRAW instead of DEPOSIT, the computer goes from line 50 to line 200 (which asks how much to withdraw), then to line 210 (which adds the 10¢ service charge to the withdrawal amount), then to line 220, which checks whether the sum in the account (S) is large enough to cover the withdrawal (W). If W<=S, the computer says OKAY, processes the check, and subtracts W from the sum in the account. If W>S instead, the computer says THAT CHECK BOUNCED and decreases the sum in the account by the $5 penalty.

How the program is nasty. That program is nasty to customers. For example, suppose you have $1 in your account, and you try to write a check for 95¢. Since 95¢ + the 10¢ service charge = $1.05, which is more than you have in your account, your check will bounce, and you'll be penalized $5, so your balance will become *negative* $4, and the bank will demand that *you* pay the *bank* $4—just because you wrote a check for 95¢!

Another nuisance is when you leave town permanently, and want to close your account. If your account contains $1, you can't get your dollar back! The most you can withdraw is 90¢, because 90¢ + the 10¢ service charge = $1.

That nasty program makes customers hate the bank—and hate the computer!

How to stop the nastiness. The bank should make the program friendlier. Here's how.

To stop accusing the customer of owing money, the bank should change any negative sum to 0:

```
15 IF S<0 THEN S=0
```

To make sure the computer goes to that line, the bank's program should say GO TO 15 instead of GO TO 20; so the bank should change line 230 to this:

```
230 PRINT "THAT CHECK BOUNCED": S=S-5: GO TO 15
```

Also, to be friendly, the bank should ignore the 10¢ service charge when deciding whether a check will clear. So the bank should eliminate line 210. On the other hand, if the check *does* clear, the bank should afterwards impose the 10¢ service charge, like this:

```
220 IF W<=S THEN PRINT "OKAY": S=S-W-.10: GO TO 15
```

So if the bank is kind, it will insert line 15, use the new version of line 230, eliminate line 210, and use the new version of line 220.

But some banks complain that those changes are *too* kind! For example, if a customer whose account contains just 1¢ writes a million-dollar check (which bounces), the new program charges him just 1¢ for the bad check; $5 might be more reasonable.

Moral: the hardest thing about programming is choosing your goal—deciding what you *want* the computer to do.

SERIES

Let's make the computer add together all the numbers from 7 to 100, so that the computer finds the sum of this series: 7+8+9+...+100. Here's how:

```
Start the Sum at 0→→→→→→→→→→→→→→→→→→→→→→→10 S=0
Make I go from 7 to 100→→→→→→→→→→→→→→→→→→20 FOR I = 7 TO 100
Increase the Sum, by adding each I to it→→30     S=S+I
                                          40 NEXT
Print the final Sum (which is 5029)→→→→→→→50 PRINT S
```

Let's make the computer add together the *squares* of all the numbers from 7 to 100, so that the computer finds the sum of this series: (7 squared) + (8 squared) + (9 squared) + . . . + (100 squared). Here's how:

```
  10 S=0
  20 FOR I = 7 TO 100
→→30     S=S+I*I
  40 NEXT
  50 PRINT S
```

It's the same as the previous program, except that line 30 says to add I*I instead of I. Line 50 prints the final sum, which is 338259.

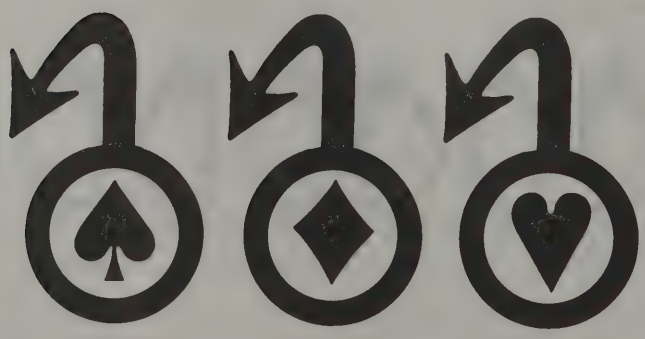
DATA SUMS

This program adds together the numbers in the data:

```
10 S=0
20 DATA 5, 3, 6.1, etc.
30 DATA etc.
40 DATA etc.
50 DATA 0
60 READ X: IF X=0 THEN PRINT S: END
70 S=S+X
80 GO TO 60
```

Line 10 starts the sum at 0. Lines 20-40 contain the numbers to be added. The zero in line 50 is an end mark.

Line 60 reads an X from the data. If X=0, the end of the data has been reached, so we want the computer to print the sum (S) and end. But if the X it reads is *not* zero, the computer proceeds from line 60 to line 70, adds X to the sum, and goes from line 80 to line 60, which reads another X.



RANDOM NUMBERS

LET'S BE UNPREDICTABLE

Usually, the computer is predictable: it does exactly what you say. But sometimes, you want the computer to be *un*predictable.

For example, if you're going to play a game of cards with the computer and tell the computer to deal, you want the cards dealt to be unpredictable. If the cards were predictable—if you could figure out exactly which cards you and the computer would be dealt—the game would be boring.

In many other games too, you want the computer to be unpredictable, to "surprise" you. Without an element of surprise, the game would be boring.

Being unpredictable increases the pleasure you derive from games—and from art. To make the computer act artistic, and create a new *original* masterpiece that's a "work of art", you need a way to make the computer get a "flash of inspiration". Flashes of inspiration aren't predictable: they're surprises.

Here's how to make the computer act unpredictably. . . .

RANDOM INTEGERS

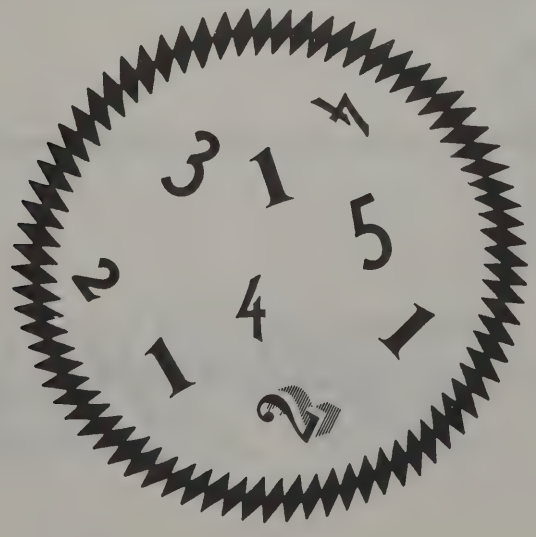
This program makes the computer print an unpredictable number from 1 to 5:

```
10 RANDOMIZE
20 PRINT RND(5)
```

Unpredictable numbers are called *random* numbers. Line 10 tells the computer to make the numbers be *completely* unpredictable, *completely* random. Line 20 makes the computer print a random number from 1 to 5, so the computer will print 1 or 2 or 3 or 4 or 5. You can't predict *which* of those numbers the computer will print. The computer's choice will be a surprise.

For example, when you type RUN, the computer might print 3. If you run the program a second time, the computer might print a different number (1 or 2 or 4 or 5), or it might print the same number (3). You can't predict which number the computer will print. The only thing you can be sure of is: the number will be from 1 to 5.

(For *your* computer, you might have to write the program slightly differently. To find out your computer's peculiarities, check the appendix.)



To make the computer print *many* such random numbers, say GO TO:

```
          10 RANDOMIZE
          20 PRINT RND(5)
New line→→30 GO TO 20
```

The computer will print many numbers, like this:

```
3
2
4
4
1
3
5
2
2
5
etc.
```

Each number will be 1 or 2 or 3 or 4 or 5. The order in which the computer prints them is unpredictable. The program's an infinite loop: it won't stop until you abort it. If you run the program again, the pattern will be different; for example, it might be:

```
1
4
3
3
2
5
1
1
2
etc.
```

When you run that program, the numbers will fly up the screen faster than you can read. To make the numbers easier to read, make the computer print them *across*, instead of down; make the computer print like this:

```
3  2  4  4  1  3  5  2  2  5  etc.
```

To do that, put a semicolon in the PRINT statement:

```
          10 RANDOMIZE
New line→→20 PRINT RND(5);
          30 GO TO 20
```

That program prints random numbers up to 5. To see random numbers up to 1000, say RND(1000):

```
          10 RANDOMIZE
New line→→20 PRINT RND(1000);
          30 GO TO 20
```

The computer will print something like this:

```
485  729  8  537  1000  13  1  842  842  156  1000  972  etc.
```

GUESSING GAME

This program plays a guessing game:

```
10 RANDOMIZE
20 PRINT "I'M THINKING OF A NUMBER FROM 1 TO 100."
30 C=RND(100)
40 INPUT "WHAT DO YOU THINK MY NUMBER IS";G
50 IF G<C THEN PRINT "YOUR GUESS IS TOO LOW.": GO TO 40
60 IF G>C THEN PRINT "YOUR GUESS IS TOO HIGH.": GO TO 40
70 PRINT "CONGRATULATIONS!  YOU FOUND MY NUMBER!"
```

Line 20 makes the computer say:

```
I'M THINKING OF A NUMBER FROM 1 TO 100.
```

Line 30 makes the computer think of a random number from 1 to 100; the computer's number is called "C". Line 40 asks the human to guess the number; the guess is called "G".

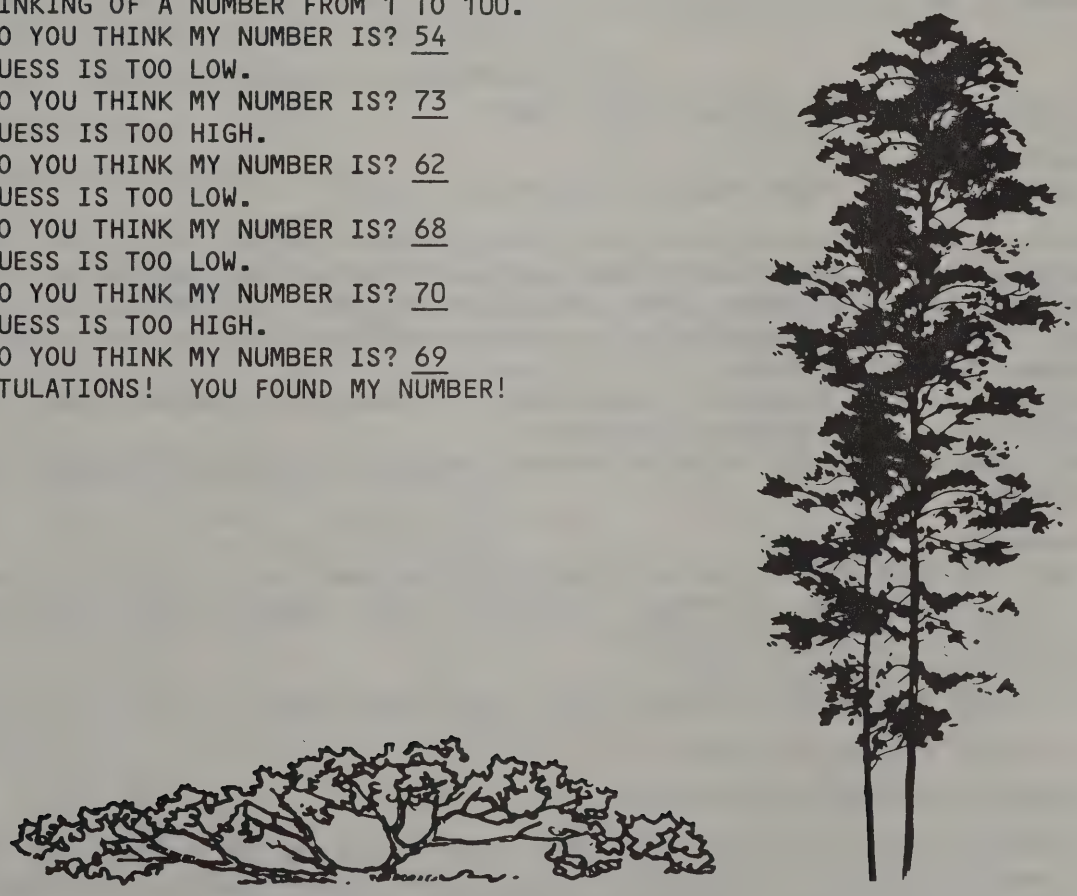
If the guess is less than the computer's number, line 50 makes the computer say "YOUR GUESS IS TOO LOW" and then GO TO 40, which lets the human guess again. If the guess is *greater* than the computer's number, line 60 makes the computer say "YOUR GUESS IS TOO HIGH" and then GO TO 40.

When the human guesses correctly, the computer arrives at line 70, which prints:

```
CONGRATULATIONS!  YOU FOUND MY NUMBER!
```

Here's a sample run:

```
RUN
I'M THINKING OF A NUMBER FROM 1 TO 100.
WHAT DO YOU THINK MY NUMBER IS? 54
YOUR GUESS IS TOO LOW.
WHAT DO YOU THINK MY NUMBER IS? 73
YOUR GUESS IS TOO HIGH.
WHAT DO YOU THINK MY NUMBER IS? 62
YOUR GUESS IS TOO LOW.
WHAT DO YOU THINK MY NUMBER IS? 68
YOUR GUESS IS TOO LOW.
WHAT DO YOU THINK MY NUMBER IS? 70
YOUR GUESS IS TOO HIGH.
WHAT DO YOU THINK MY NUMBER IS? 69
CONGRATULATIONS!  YOU FOUND MY NUMBER!
```



DICE

This program makes the computer roll a pair of dice:

```
10 RANDOMIZE
20 PRINT "I'M ROLLING A PAIR OF DICE"
30 A=RND(6)
40 PRINT "ONE OF THE DICE SAYS";A
50 B=RND(6)
60 PRINT "THE OTHER SAYS";B
70 PRINT "THE TOTAL IS";A+B
```

Line 20 makes the computer say:

```
I'M ROLLING A PAIR OF DICE
```

Each of the dice has 6 sides. Lines 30 and 40 roll one of the dice, by picking a number from 1 to 6. Lines 50 and 60 roll the other. Line 70 prints the total.

Here's a sample run:

```
I'M ROLLING A PAIR OF DICE
ONE OF THE DICE SAYS 3
THE OTHER SAYS 5
THE TOTAL IS 8
```

Here's another run:

```
I'M ROLLING A PAIR OF DICE
ONE OF THE DICE SAYS 6
THE OTHER SAYS 4
THE TOTAL IS 10
```

COIN FLIPPING

This program makes the computer flip a coin:

```
10 RANDOMIZE
20 IF RND(2)=1 THEN PRINT "HEADS" ELSE PRINT "TAILS"
```

RND(2) is a random number from 1 to 2; so it's either 1 or 2. If it's 1, line 20 makes the computer say HEADS; if it's 2 instead, line 20 makes the computer say TAILS.

Until you type RUN, you won't know which way the coin will flip; the choice is random. Each time you type RUN, the computer will flip the coin again; each time, the outcome is unpredictable.

(Warning: if your computer's too stupid to understand the word ELSE, you must retype line 20; the appendix explains how.)

Let's permit the human to bet on whether the computer will say HEADS or TAILS. Here's how:

```
  10 RANDOMIZE
→→15 INPUT "DO YOU WANT TO BET ON HEADS OR TAILS";B$
→→16 IF B$<>"HEADS" AND B$<>"TAILS" THEN PRINT "SAY HEADS OR TAILS": GO TO 15
→→20 IF RND(2)=1 THEN C$="HEADS" ELSE C$="TAILS"
→→30 PRINT "THE COIN SAYS ";C$
→→40 IF C$=B$ THEN PRINT "YOU WIN" ELSE PRINT "YOU LOSE"
```

Line 15 makes the computer ask:

```
DO YOU WANT TO BET ON HEADS OR TAILS?
```

Line 16 makes sure the human says HEADS or TAILS: if the human's answer isn't HEADS and isn't TAILS, the computer gripes. Lines 20 and 30 make the computer flip a coin. Line 40 determines whether the human won or lost the bet.

Here's a sample run:

```
DO YOU WANT TO BET ON HEADS OR TAILS? HEADS
THE COIN SAYS TAILS
YOU LOSE
```

Here's another:

```
DO YOU WANT TO BET ON HEADS OR TAILS? TAILS
THE COIN SAYS TAILS
YOU WIN
```

Here's another:

```
DO YOU WANT TO BET ON HEADS OR TAILS? TAILS
THE COIN SAYS HEADS
YOU LOSE
```

To make the program more fun, let the human use money when betting:

```
→→1 RANDOMIZE
→→2 S=100
→→3 PRINT "YOU HAVE";S;"DOLLARS"
→→10 INPUT "HOW MANY DOLLARS DO YOU WANT TO BET";B
→→11 IF B>S THEN PRINT "YOU DON'T HAVE THAT MUCH!  YOU MUST BET LESS!": GO TO 10
→→12 IF B<0 THEN PRINT "YOU CAN'T BET LESS THAN NOTHING!": GO TO 10
→→13 IF B=0 THEN PRINT "I GUESS YOU DON'T WANT TO BET ANYMORE": GO TO 100
  15 INPUT "DO YOU WANT TO BET ON HEADS OR TAILS";B$
  16 IF B$<>"HEADS" AND B$<>"TAILS" THEN PRINT "SAY HEADS OR TAILS": GO TO 15
  20 IF RND(2)=1 THEN C$="HEADS" ELSE C$="TAILS"
  30 PRINT "THE COIN SAYS ";C$
→→40 IF C$=B$ THEN PRINT "YOU WIN";B;"DOLLARS": S=S+B: GO TO 3
→→50 PRINT "YOU LOSE";B;"DOLLARS": S=S-B: IF S>0 THEN GO TO 3
→→60 PRINT "YOU'RE BROKE!  TOO BAD!"
→→100 PRINT "THANKS FOR PLAYING WITH ME!  YOU WERE FUN TO PLAY WITH!"
→→110 PRINT "I HOPE YOU PLAY AGAIN SOMETIME"
```

Lines 2 and 3 make the computer say:

```
YOU HAVE 100 DOLLARS
```

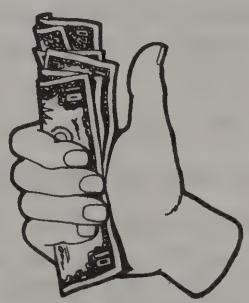
Line 10 makes the computer ask:

```
HOW MANY DOLLARS DO YOU WANT TO BET?
```

Lines 11-13 make sure the bet is reasonable. Lines 15 and 16 get the human to bet on heads or tails. Lines 20 and 30 flip the coin. Lines 40 and 50 determine whether the human won or lost the bet, and then send the computer back to line 3 for another round (if the human isn't broke yet). Lines 60-110 say good-bye to the human.

Here's a sample run:

```
YOU HAVE 100 DOLLARS
HOW MANY DOLLARS DO YOU WANT TO BET? 120
YOU DON'T HAVE THAT MUCH!  YOU MUST BET LESS!
HOW MANY DOLLARS DO YOU WANT TO BET? 75
DO YOU WANT TO BET ON HEADS OR TAILS? HEADS
THE COIN SAYS TAILS
YOU LOSE 75 DOLLARS
YOU HAVE 25 DOLLARS
HOW MANY DOLLARS DO YOU WANT TO BET? 10
DO YOU WANT TO BET ON HEADS OR TAILS? TAILS
THE COIN SAYS TAILS
YOU WIN 10 DOLLARS
YOU HAVE 35 DOLLARS
HOW MANY DOLLARS DO YOU WANT TO BET? 35
DO YOU WANT TO BET ON HEADS OR TAILS? TAILS
THE COIN SAYS HEADS
YOU LOSE 35 DOLLARS
YOU'RE BROKE!  TOO BAD!
THANKS FOR PLAYING WITH ME!  YOU WERE FUN TO PLAY WITH!
I HOPE YOU PLAY AGAIN SOMETIME
```



To make the output prettier, insert these lines:

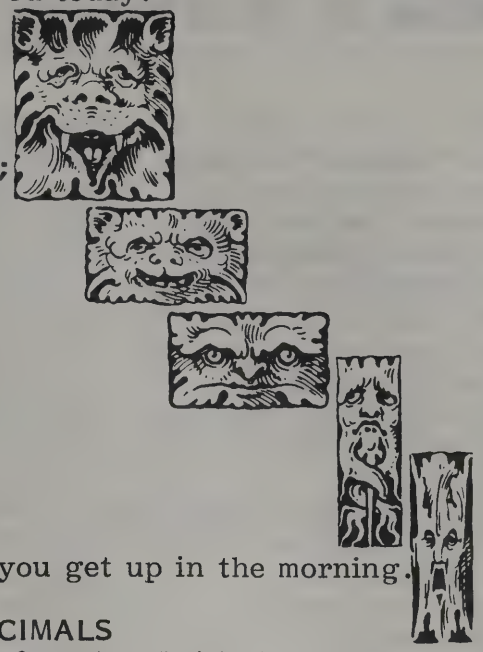
```
3 PRINT
4 PRINT "YOU HAVE";S;"DOLLARS!  HERE THEY ARE:"
5 FOR I = 1 TO S
6     PRINT "$";
7 NEXT
8 PRINT
```

Now the run looks like this:

```
RUN

YOU HAVE 100 DOLLARS!  HERE THEY ARE:
$$$
```

DAILY HOROSCOPE

This program predicts what will happen to you today:



```
10 RANDOMIZE
20 PRINT "YOU WILL HAVE A ";
30 R=RND(5)
40 IF R=1 THEN PRINT "WONDERFUL";
50 IF R=2 THEN PRINT "BETTER-THAN-AVERAGE";
60 IF R=3 THEN PRINT "SO-SO";
70 IF R=4 THEN PRINT "WORSE-THAN-AVERAGE";
80 IF R=5 THEN PRINT "TERRIBLE";
90 PRINT " DAY TODAY"
```

The computer will say—

```
YOU WILL HAVE A WONDERFUL DAY TODAY
```

or—

```
YOU WILL HAVE A TERRIBLE DAY TODAY
```

or some in-between comment.

For inspiration, run that program when you get up in the morning.

RANDOM DECIMALS

You've seen that RND(5) is a random number from 1 to 5: it's 1 or 2 or 3 or 4 or 5. To get a random *decimal* between 0 and 1, say just RND instead of RND(5).

The decimal that RND produces is at least 0 and is less than 1, so it can be any decimal from 0.0000000 to 0.9999999. For example, the decimal might be 0.2845918.

Suppose you want the computer to *maybe* print LOVE. Here's how to make the probability of printing LOVE be 37 percent:

```
10 RANDOMIZE
20 IF RND<.37 THEN PRINT "LOVE"
```

.37

WOW! YOU FINISHED CHAPTER 5: POWERFUL PROGRAMMING!
TURN THE PAGE FOR CHAPTER 6: DOS.

GETTING INTO DOS

IBM PC SOFTWARE

The most popular computer for business applications is the IBM PC. The best business software runs only on the IBM PC and its clones.

Someday, better software might become available for other computers. For example, many programmers are trying to develop high-quality business software for the Macintosh. But business software for the Macintosh and other computers isn't yet as good as the software for the IBM PC. So if you want to buy a computer for your business, I recommend that you buy an IBM PC or a clone—unless your business is very unusual!

Software for the IBM PC comes on floppy disks. When you buy the floppy disks, you also get instruction manuals that explain how to use the disks and run their software.

The typical instruction manual assumes that your computer system is made by IBM and includes two double-sided floppy-disk drives, at least 256K of RAM, a parallel printer port, a printer attached to that port, and an RGB color monitor. If your computer system's slightly different, you'll face difficulties, though the instruction manuals will try to help you overcome them.

YOUR FIRST DISK

For your IBM PC, the first disk of software to buy is called the *DOS disk*. More precisely, it's called the *PC-DOS System Master disk*.

The software on that disk teaches the computer how to handle the disk drives. You must insert that disk into the drive before you use any other disks.

VERSIONS OF PC-DOS

The original version of PC-DOS was called *version 1.0*. Then came an improvement called *version 1.1*. Then came version 2.0, then 2.1, then 3.0, and finally 3.1.

Versions 1.0 and 1.1 handled just the original IBM PC and its double-density floppy disks. Versions 2.0 and 2.1 could also handle the IBM PC XT and its hard disk. Versions 3.0 and 3.1 could also handle the IBM PC AT and its high-density floppy disks.

Although version 3.1 is the fanciest, it has two disadvantages: it consumes lots of RAM, and it can't handle some older software. Most business software was written for version 2.1. So buy version 2.1, unless you have an IBM PC AT (which *requires* version 3.1). Versions 2.1 and 3.1 each cost $65.

HOW TO START USING PC-DOS

Here's how to use the most popular version (2.1) on a standard IBM PC. . . .

A standard IBM PC has two floppy-disk drives. The main drive, which is on the left, is called *drive A*. The other floppy-disk drive, which is on the right, is called *drive B*.

(If you have a PC Junior instead, you get just one drive, which is called A. If you have a PC XT or PC AT instead, you typically get one floppy drive, called A, and one hard-disk drive, called C.)

Put the PC-DOS System Master disk into drive A. (The disk's label says "DOS" and the version number. When putting the disk into the drive, make sure the disk's label is on *top* of the disk, and make sure the disk's big oval cutout goes into the drive *before* the label does. After putting the disk into the drive, close the drive's door.)

Turn on the TV or monitor. (If you're using a TV, turn it to channel 3 or 4 or 33.) Then flip the computer's power switch to the ON position. (On the standard IBM PC, the power switch is red and at the right side.) If you put your ear next to the computer, you'll hear the computer's fan hum softly. After 8 seconds, the screen becomes black, except for one blinking horizontal line in the upper left corner.

About 45 seconds later (the exact time depends on how much RAM you own), the computer will beep and say:

```
Enter new date:
```

Type the date. For example, if today is January 27, 1986, type 1-27-86 (and then press the ENTER key, which is a big key on the right side and has a bent arrow on it).

The computer will say:

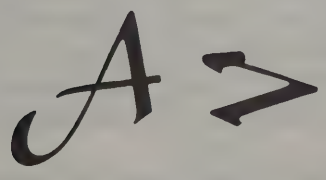
```
Enter new time:
```

Type the time, using a 24-hour clock. For example, if the time is 1:45 AM, type 1:45. If the time is 1:45 PM, type 13:45. If the time is 7 seconds after 1:45 PM, type 13:45:07. (After typing the time, press the ENTER key.)

The computer will say:

```
A>
```

That "A>" is called the *A prompt*. Notice that the A prompt is typed by the computer, not by you!



After the computer types the A prompt, you can type many kinds of commands, which are called *PC-DOS commands*.

For example, after the A prompt you can type "dir", like this:

```
A>dir
```

The "dir" command makes the computer print a *directory* of the files that are stored on the disk in drive A. The computer prints the directory onto your screen. The directory looks like this:

```
COMMAND  COM    17792  10-20-83  12:00p
ANSI     SYS     1664  10-20-83  12:00p
FORMAT   COM     6912  10-20-83  12:00p
CHKDSK   COM     6400  10-20-83  12:00p
SYS      COM     1680  10-20-83  12:00p
DISKCOPY COM     2576  10-20-83  12:00p
DISKCOMP COM     2188  10-20-83  12:00p
COMP     COM     2534  10-20-83  12:00p
EDLIN    COM     4608  10-20-83  12:00p
MODE     COM     3139  10-20-83  12:00p
FDISK    COM     6369  10-20-83  12:00p
BACKUP   COM     3687  10-20-83  12:00p
RESTORE  COM     4003  10-20-83  12:00p
PRINT    COM     4608  10-20-83  12:00p
RECOVER  COM     2304  10-20-83  12:00p
ASSIGN   COM      896  10-20-83  12:00p
TREE     COM     1513  10-20-83  12:00p
GRAPHICS COM      789  10-20-83  12:00p
SORT     EXE     1408  10-20-83  12:00p
FIND     EXE     5888  10-20-83  12:00p
MORE     COM      384  10-20-83  12:00p
BASIC    COM    16256  10-20-83  12:00p
BASICA   COM    26112  10-20-83  12:00p
       23 File(s)     28672 bytes free
```

The first line of that directory says the disk in drive A contains a file whose name is "COMMAND.COM", which contains 17792 bytes and was last updated on October 20, 1983, at 12:00PM. The next line says the disk also contains a file whose name is "ANSI.SYS", which contains 1664 bytes and was updated on October 20, 1983 also. The remaining lines say that the disk also contains a file called "FORMAT.COM", a file called "CHKDSK.COM", and other files. Altogether, the directory names the 23 typical files that are on the disk.

In addition to those 23 typical files, the disk also contains 7 special items. When the computer prints the directory, it doesn't bother to mention those 7 special items, because practically *every* IBM disk contains them. (Those 7 special items are the "IBMIO.COM" file, the "IBMDOS.COM" file, the "label file", the "boot record", the "first file allocation table", the "second file allocation table", and the "directory itself".)

The 23 typical files, together with the 7 special items, consume most of the usable surface of the disk, but the disk still has enough room left to hold 28,672 bytes of additional data—as indicated by the directory's bottom line.

Above that directory, the computer also prints a message saying "volume in drive A has no label". That means the disk is simple, and doesn't contain an extra label.

Abridgments. If you type "dir command.com", like this—

```
A>dir command.com
```

the computer will print an abridged directory, that shows just the information about information about the file called COMMAND.COM. The computer will print just:

```
COMMAND  COM    17792  10-20-83  12:00p
       1 File(s)      28672 bytes free
```

If you type "dir *.exe", the computer will print an abridged directory, showing information about just the files whose names end in ".exe"; the computer will print:

```
SORT     EXE     1408  10-20-83  12:00p
FIND     EXE     5888  10-20-83  12:00p
       2 File(s)      28672 bytes free
```

The symbol "*" means "anything". That's why, when you type "dir *.exe", the computer shows a directory of anything that ends in ".exe".

If you type "dir *.com", you'll see a directory of anything that ends in ".com".

If you type "dir f*.com", you'll see a directory of files whose names begin with "f" and end with ".com"; the computer will print:

```
FORMAT   COM     6912  10-20-83  12:00p
FDISK    COM     6369  10-20-83  12:00p
       2 File(s)      28672 bytes free
```

If you type "dir f*.*", you'll see a directory of files whose names begin with "f" and end with anything; the computer will print:

```
FORMAT   COM     6912  10-20-83  12:00p
FDISK    COM     6369  10-20-83  12:00p
FIND     EXE     5888  10-20-83  12:00p
       3 File(s)      28672 bytes free
```

If you type "dir ba*.*", you'll see a directory of files whose names begin with "ba" and end with anything; the computer will print:

```
BACKUP   COM     3687  10-20-83  12:00p
BASIC    COM    16256  10-20-83  12:00p
BASICA   COM    26112  10-20-83  12:00p
       3 File(s)      28672 bytes free
```

If you type "dir/w", you'll see a directory that's wide and leaves out the details; the computer will print:

```
COMMAND  COM    ANSI     SYS    FORMAT   COM    CHKDSK   COM    SYS      COM
DISKCOPY COM    DISKCOMP COM    COMP     COM    EDLIN    COM    MODE     COM
FDISK    COM    BACKUP   COM    RESTORE  COM    PRINT    COM    RECOVER  COM
ASSIGN   COM    TREE     COM    GRAPHICS COM    SORT     EXE    FIND     EXE
MORE     COM    BASIC    COM    BASICA   COM
       23 File(s)      28672 bytes free
```

EXTERNAL COMMANDS

LET'S EXPLORE

The System Master contains 23 files. Why? To find out, let's try using some of them. . . .

SORT

One of the files on the System Master is "SORT.EXE". It alphabetizes. To use it, type "sort" after the A prompt, like this:

```
A>sort
```

Underneath, type a list of words, such as:

```
pastrami
salami
jelly
belly
rose
toes
```

After typing those words (and pressing the ENTER key after toes), press the F6 key, which is on the left side of the keyboard. (On the PC Junior, which lacks an F6 key, press the Fn key and then press the 6 key.) After pressing the F6 key (or equivalent), press the ENTER key. The computer will sort the words and print them in alphabetical order; it will print:

```
belly
jelly
pastrami
rose
salami
toes
```

CHKDSK

One of the files on the System Master is "CHKDSK.COM". That file CHecKs your DiSK and also your RAM. To use it, type "chkdsk" after the A prompt, like this:

```
A>chkdsk
```

It makes the computer print a message such as:

```
179712 bytes total disk space
 22528 bytes in 3 hidden files
128512 bytes in 23 user files
 28672 bytes available on disk

262144 bytes total memory
237568 bytes free
```

That means 179712 bytes on the disk can be used for holding files (the remaining bytes on the disk are used for holding some of the special items). Of those 179712 bytes, 128512 bytes are used by the 23 files that the directory mentioned and by the blank spaces marking the end of each of those files, 22528 bytes are used by the 3 files that the directory didn't mention ("IBMIO.COM", "IBMDOS.COM", and the label file), and the remaining 28672 bytes aren't used yet and are still available for any files you wish to add.

The message also says your computer's RAM is large enough to hold 262144 bytes altogether. A few of those bytes are used by the DOS, but the message says 237568 bytes remain unused. The unused bytes are available for any programs you wish to run.

ABSENT FILES

Notice that the System Master does *not* contain a file named "PUSHUPS.COM". If you try to make the computer do pushups, by saying—

```
A>pushups
```

the computer will look on the disk for a file named "PUSHUPS.COM", not find it, and then gripe, by saying:

```
Bad command or file name
```

VERSIONS OF BASIC

One file on the System Master is called "BASIC.COM". To use it, type "basic", like this:

```
A>basic
```

It makes the computer say "Ok" and let you write programs in BASIC. When you've finished writing programs in BASIC, type the word "system" (and press the ENTER key); the computer will print the A prompt again.

Another file on the System Master is called "BASICA.COM". To use it, type "basica", like this:

```
A>basica
```

It makes the computer say "Ok" and let you write programs using a more advanced version of BASIC. When you've finished writing programs in advanced BASIC, type the word "system" to make the computer print the A prompt.

Warning: the versions of "BASIC.COM" and "BASICA.COM" that come on the PC-DOS System Master work only if your computer is manufactured by IBM. If your computer is a clone (an imitation manufactured by a different company), "BASIC.COM" and "BASICA.COM" won't work. To use BASIC on a clone, you must use an *MS-DOS System Master Disk* instead of a PC-DOS System Master Disk. MS-DOS is almost the same as PC-DOS, but designed especially for clones. The version of BASIC on the MS-DOS System Master Disk *will* work on your clone. That version of BASIC is called "BASIC.COM" or "GWBASIC.COM", and is about as good as IBM's "BASICA.COM".

DISKCOPY

One file on the System Master is "DISKCOPY.COM". It makes copies of disks. To use it, type "diskcopy a: b:", so that your screen looks like this:

```
A>diskcopy a: b:
```

The computer will say:

```
Insert source diskette in drive A:
Insert target diskette in drive B:
Strike any key when ready
```

That means, "put the disk you want to copy into drive A, put a blank disk into drive B, and then strike any key on the keyboard."

Do it! Make sure the disk you put into drive B is blank: if you make a mistake and insert a disk that's *not* blank, all information on that disk will be destroyed! After you've put the disks into the drives (and closed the drive doors), strike any key on the keyboard.

Then the computer will copy all the information from drive A's disk to drive B's disk, so that drive B's disk becomes an exact copy of drive A's. The copying takes about a minute. (The exact time depends on how much information was on the disk).

When the copying is finished, the computer asks:

```
Copy another (Y/N)?
```

If you want to copy another disk, tap the Y key (which means "Yes"); otherwise, tap the N key (which means "No").

In that procedure, I assumed you had a second floppy-disk drive, called B. If you have just *one* floppy-disk drive, here's what to do. . . .

Instead of saying "diskcopy a: b:", say "diskcopy a: a:". The computer will say:

```
Insert source diskette in drive A:
Strike any key when ready
```

Put the disk you want to copy into drive A, then tap any key on the keyboard. The computer will copy information from that disk to the RAM. Then the computer will tell you to insert a blank disk. Then the computer will copy the information from the RAM to the blank disk. If your RAM is too small to hold all the disk's information, the computer will copy just *part* of the disk's information into the RAM, then copy that part onto the blank disk, then automatically go through the whole procedure again to copy the remaining information.

When you buy a new IBM computer, the first thing you should do is make a copy of the System Master disk, by saying "diskcopy a: b:" (or "diskcopy a: a:"). Then use the copy. Store the original disk in a safe place far away, so that if you ever accidentally damage the copy you can go back to the original.

Although the "diskcopy a: b:" command usually works, sometimes it doesn't! The computer might *refuse* to copy a disk! That happens if you buy a disk whose programs were written by nervous programmers, who fear you'll copy the disk for your friends without paying royalties. Those programmers alter the disk, to prevent "diskcopy" from working.

FORMAT

Suppose you buy a blank disk and want to use it. One way to use it is to make it become a copy of another disk, by putting it into drive B and saying "diskcopy a: b:". But suppose you don't want the blank disk to be a copy of another disk; suppose you want to use it to store different files instead. To do that, you must *format* the blank disk, by using a program on the System Master called "FORMAT.COM". Here's how. . . .

Put the System Master disk (or a copy of it) into drive A and type "format/s b:", like this:

```
A>format/s b:
```

The computer will say:

```
Insert new diskette for drive B:
and strike any key when ready
```

If your computer doesn't understand "format/s b:", say "format b:/s" instead.

Into drive B, put the blank disk that you want to format. Then press any key. The computer will format the disk in drive B. The formatting takes 1¼ minutes. Then the computer will ask:

```
Format another (Y/N)?
```

If you want to format another blank disk, press the Y key; otherwise, press N.

If your computer doesn't have a drive B, say "format/s a:", and put the blank disk into drive A instead of B.

Warnings. Make sure the disk you're formatting was blank. If it wasn't blank, the computer will erase all the information that was on it!

If your computer has a hard disk (which is called "C"), do *not* say "format/s c:", since that command would make the computer erase all the information that was on your hard disk!

Make sure you say which disk to format. If you want to format disk A, say "format/s a:". If you want to format disk B, say "format/s b:". If you want to format disk C, say "format/s c:".

If you don't say which disk to format, and just say "format/s", the computer will format whichever disk it wishes, which might not be the disk you intended!

The typical business that buys a hard disk will put all the valuable information about the business onto the hard disk. But then, 3 months later, all the information on the hard disk will be accidentally erased, and the business will be destroyed! Here's why the hard disk gets accidentally erased. . . .

About 3 months after buying a hard disk, untrained employees start trying to format their own floppy disks. One of those employees, who meant to say "format/s a:", forgets to type the "a:", and says just "format/s". Since the employee forgot to say which disk to format, the computer chooses whichever disk it wishes—which might be disk C, the hard disk! So the computer erases the entire hard disk! That's the quickest way to lose 3 months of work!

Then the company's auditors have lots of fun. When they ask the company, "Where are your financial records for the past quarter?", the company will have to say, "Completely gone!"

What the "/s" means. In the format command, the "/s" tells the computer to put onto the disk these three files: "COMMAND.COM", "IBMIO.COM", and "IBMDOS.COM". Those three files make the disk work well in any drive. The "COMMAND.COM" file will appear in the disk's directory; the other two files will be hidden.

If you omit the "/s", and say just "format b:" instead of "format/s b:", the computer will *not* put "COMMAND.COM", "IBMIO.COM", and "IBMDOS.COM" onto the disk, and the disk will *not* work properly in drive A, though you can still use it in drive B.

A disk that was formatted by saying "format/s b:" is called a *system disk* (or *bootable disk*). The "/s" means: make it a system disk.

A disk that was formatted by saying just "format b:" is called a *non-system disk* (or *data disk*).

A system disk works well in either drive. A non-system disk works well only in drive B.

Volume label. While you're formatting a disk, you can also give it a name, which is called the *volume label*. To do that, add "/v" to the format command, like this:

```
A>format/s/v b:
```

The computer will format the disk, and then it will ask:

```
Volume label (11 characters)?
```

To reply, invent a name (volume label) for the disk, and type the name; then press the ENTER key. Keep the name short: no more than 11 characters.

In the future, whenever you ask the computer to print the disk's directory, the computer will automatically print the disk's name at the top of the directory.

INTERNAL COMMANDS

INTERNAL VERSUS EXTERNAL

To use commands such as "sort", "chkdsk", "basic", "basica", "diskcopy a: b:", and "format/s b:", you must put the System Master into drive A, because those commands use the files on the System Master disk ("SORT.EXE", "CHKDSK.COM", "BASIC.COM", "BASICA.COM", "DISKCOPY.COM", and "FORMAT.COM").

But the "dir" command does *not* require that drive A contain a System Master. The "dir" command works even if drive A contains just a plain system disk (created by using "format/s b:"), because even a plain system disk contains the definition of "dir" on it; and that definition is automatically copied from the system disk to the RAM when you turn on the computer.

Since the definition of "dir" stays in the RAM permanently (or until you turn off the computer's electricity), "dir" is called an *internal command*. Commands such as "sort", "chkdsk", "basic", "basica", "diskcopy", and "format", whose definitions are only on the System Master and don't stay in the RAM, are called *external commands*.

Here's the rule: an external command works only if drive A contains the System Master; an internal command works even if drive A contains just a plain system disk.

POPULAR INTERNAL COMMANDS

The most popular internal commands are "dir", "copy", "type", "rename", "erase", and the colon commands. Let's examine them.

Suppose you format two disks (by saying "format/s b:"), and then put one of the disks into drive A and the other into drive B. Let's have some fun with them.

The Jewish religion prohibits Orthodox Jews from eating ham. That's why Mary had a little lamb:

Mary had a little lamb,
'Cause Jewish girls can't eat no ham.
If Mary were a Hindu now,
Mary couldn't eat no cow.
Religions all are fine and dandy,
Even my dentist who says, "No candy!"
But Ma's religion makes me shiver.
That's why mine says, "Ma, no liver!"

Here's how to put that poem onto drive A's disk and call it MARY. First, type "copy con mary", like this:

```
A>copy con mary
```

Underneath that line, type the poem. (If you don't like that poem, make up your own!) Underneath your poem, press the F6 key and then the ENTER key. The computer will automatically copy your poem onto the disk and call it MARY.

To prove that your computer put the poem onto the disk, look at the disk's directory, by typing "dir". You'll see that one of the files in the directory is MARY.

If you say "type mary", the computer will type the entire poem on your screen. (Do *not* try to make the computer type a file that ends in ".com" or ".exe"; those files consists of commands or executions that can't be typed.)

MARY is in drive A. To change its name to LAMBCHOP, say "rename mary lambchop". To copy LAMBCHOP to the disk in drive B, say "copy lambchop b:". To prove that it's in drive B, make the computer print the directory of drive B, by saying "dir b:".

You can give a file any short name you wish, such as MARY or LAMBCHOP. Keep the filename short: you can't make it longer than 8 characters.

To erase LAMBCHOP from drive A, say "erase lambchop". To erase LAMBCHOP from drive B, say "erase b:lambchop".

CHANGING THE MAIN DRIVE

Usually, drive A is the main drive, and the computer prints "A>" on the screen. If you want drive B to be the main drive instead, type "b:" after the "A>", so that your screen looks like this:

```
A>b:
```

When you press the ENTER key at the end of that line, the computer will make drive B become the main drive. The computer will say:

```
B>
```

If you have a hard drive (called C), you can make that drive become the main drive, by typing "c:". The computer will say:

```
C>
```

If you say "dir a:", the computer prints a directory of drive A. If you say "dir b:", the computer prints a directory of drive B. If you say "dir c:", the computer prints a directory of drive C. If you say just "dir", without specifying which drive, the computer prints the directory of the main drive (which is drive A if the computer said "A>", drive B if the computer said "B>", and drive C if the computer said "C>").

SUMMARY OF INTERNAL COMMANDS

Here's a summary of all popular internal commands:

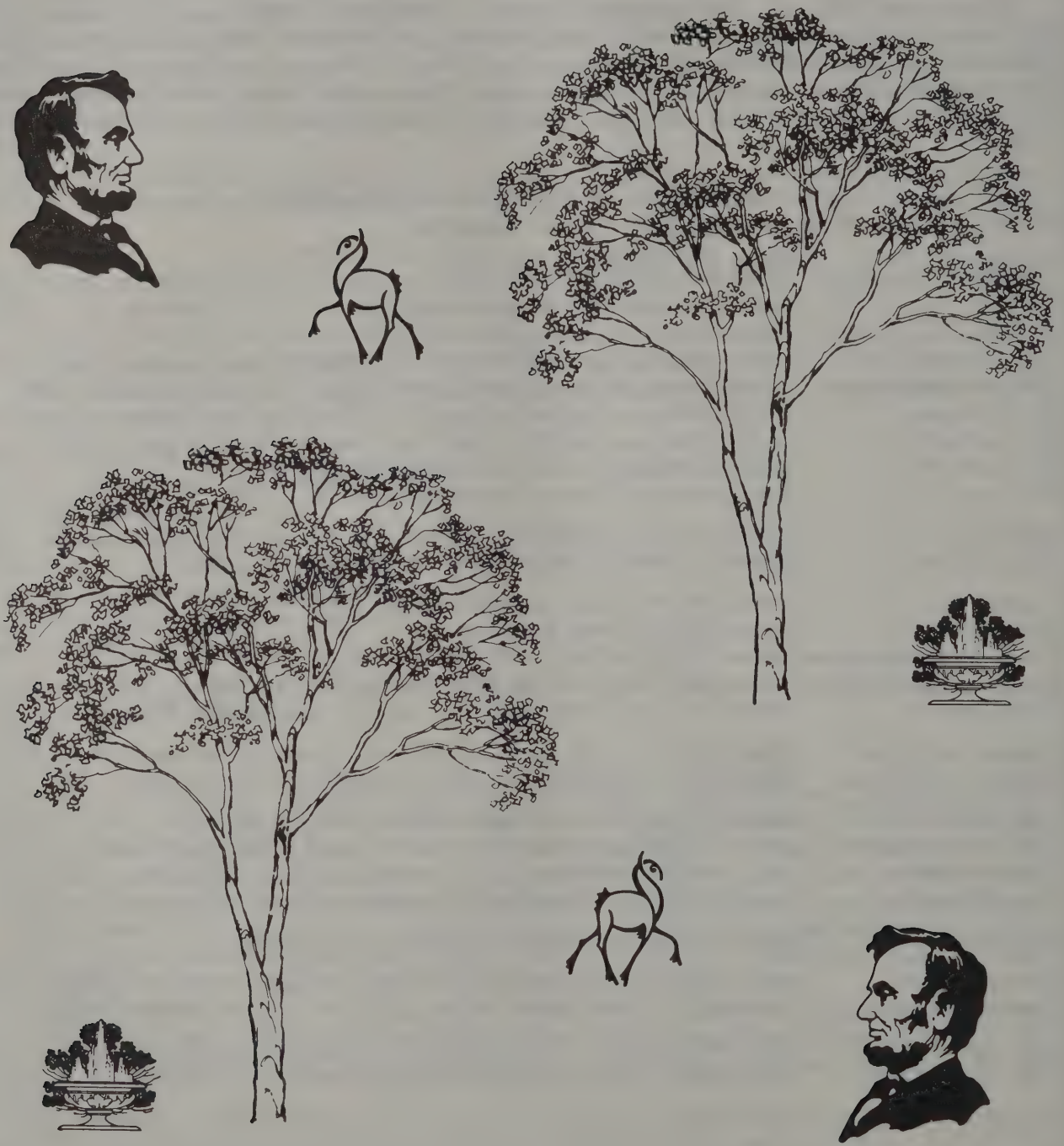
Command	What the computer will do
`dir`	print a directory of the main drive
`dir b:`	print a directory of drive B
`dir b:mary`	print an abridged directory of drive B, showing just MARY
`dir b:command.com`	print abridged directory showing just B's "COMMAND.COM"
`dir b:*.com`	print directory showing just B's files that end with ".COM"
`dir b:f*.com`	print directory of B's files that begin with F & say ".COM"
`dir b:f*.*`	print directory of B's files that begin with F
`dir/w b:`	print wide directory of drive B
`type mary`	type all lines of MARY, which is a file on the main drive
`type b:mary`	type all lines of MARY, which is a file on drive B
`erase mary`	erase MARY from the main drive
`erase b:mary`	erase MARY from drive B
`erase b:command.com`	erase "COMMAND.COM" from drive B
`erase b:*.com`	erase from drive B all files ending in ".COM"
`erase b:*.*`	erase all files from drive B (except the hidden files)
`rename mary sue`	find MARY on the main drive, and change the name to SUE
`rename b:mary sue`	find MARY on drive B, and change the name to SUE
`copy mary b:`	copy MARY from the main drive to drive B
`copy command.com b:`	copy "COMMAND.COM" from the main drive to drive B
`copy *.com b:`	copy all files ending in ".COM" from the main drive to B
`copy *.* b:`	copy all visible files from the main drive to drive B
`copy b:mary`	copy MARY from drive B to the main drive
`copy mary sue`	copy MARY; call the copy "SUE"; keep both on main drive
`copy mary b:sue`	copy MARY from main drive to B, but call B's copy "SUE"
`copy con mary`	copy MARY from the console (keyboard) to the main drive
`copy con b:mary`	copy MARY from the console (keyboard) to drive B

TWO WAYS TO COPY A DISK

You can copy files from an original disk to a blank disk in two ways.

The fastest way is to put the System Master in drive A, type "diskcopy a: b;", and then—when the computer asks you to put the disks in the drives—put the original disk into drive A and the blank disk into drive B.

A slower but more sophisticated way is to put the System Master in drive A, put the blank disk in drive B, format the blank disk (by typing "format/s b:"), then put the original disk into drive A and type "copy *.* b:". Though that way's slow, it has one advantage: while the computer performs the "copy *.* b:", the computer analyzes the files it's copying and organizes them better on the new disk (B) than they were originally (on A). Because the files on the new disk are organized better, the computer will be able to find and use them faster than the files on the original.



ADVANCED FEATURES

HOW TO USE THE PRINTER

Normally, the computer prints its answers on the screen. To make the computer print its answers on paper instead (by using the printer), type ">prn" at the end of the command.

For example, to make the computer send a directory of drive B to the printer (instead of to your screen), say "dir b: >prn". To make the printer type all the lines of file MARY onto the printer (instead of onto your screen), say "type mary >prn".

Another way to copy all the lines of MARY onto your printer's paper is to simply say "copy mary prn".

To send information directly from your keyboard (console) to the printer, say "copy con prn"; underneath that command, type whatever poetry you want to send to the printer; and underneath your poem, press the F6 key and then the ENTER key. Your printer will print the entire poem.

Two other ways to send information to the printer are to get a screen dump (by holding down the SHIFT and PrtSc keys simultaneously) or to echo (by holding down the Ctrl and PrtSc keys simultaneously, then putting onto the screen whatever you want the computer to print). To turn the echo off, hold down the Ctrl and PrtSc keys simultaneously again.

BATCH FILES

You can invent your own command, and make it stand for a list of other commands.

For example, suppose you want to invent a command called "both", which will make the computer give you a directory of both drives (A and B). To do that, create a file called "BOTH.BAT", which contains two lines, "dir" and "dir b:". To create such a file, type as follows—

```
A>copy con both.bat
dir
dir b:
```

then press the F6 key and then the ENTER key.

Afterwords, whenever you type the word "both", like this—

```
A>both
```

the computer will look at the file "BOTH.BAT" and obey the commands you stored there: the computer will automatically do "dir" and then "dir b:".

A file that's a list of commands is called a *batch file*. The file "BOTH.BAT" is a batch file, because it's a list of two commands ("dir" and "dir b:"). The name of every batch file must end in ".BAT", which stands for "batch".

For another example, let's invent a new command called "status", which makes the computer tell you the status of drive A and the RAM. So we'll define "status" to mean "dir" followed by "chkdsk". Type this—

```
A>copy con status.bat
dir
chkdsk
```

then press the F6 and ENTER keys. Then whenever you type "status", like this—

```
A>status
```

the computer should automatically do "dir" and "chkdsk".

For that new "status" command to work, drive A must contain "STATUS.BAT" and "CHKDSK.COM". If you try to say "status" when drive A does *not* contain "STATUS.BAT" and "CHKDSK.COM", the computer won't know what you're talking about, so it will gripe.

BOOT-UP DETAILS

When you turn on the computer, it automatically goes through a procedure called the *bootstrap*. Here's how it works.

First, the computer tests its own RAM, to make sure the RAM is working properly. (If the RAM is defective, the computer prints a number on the screen.)

Next, the computer tests the keyboard cable, to make sure it's properly attached to the computer. (If the keyboard cable is defective—or not plugged in—the computer prints the number "301" on the screen, if your computer is an IBM PC. If you have a clone instead, your computer might print a different number or comment.)

Next, the computer decides which drive is the "main" one. To decide, the computer begins by checking whether a disk is in drive A. If drive A contains a disk, drive A becomes the main drive, and the computer prints this on your screen:

```
A>
```

If drive A does *not* contain a disk (or the drive's door is accidentally open), the computer checks whether you own a hard-disk drive. If you own one, the computer calls it "C", makes it the main drive, and prints this on your screen:

```
C>
```

Since C is the main drive, "dir" will make the computer give you a directory of drive C (instead of drive A).

If drive A doesn't contain a disk, but you don't have a hard-disk drive either, the computer gives up, ignores the whole concept of PC-DOS, and says:

```
The IBM Personal Computer Basic
```

In that situation, you can immediately start writing programs in BASIC, but you can't give any DOS commands, and you can't use any disks. (Exception: if you have a clone instead of a real IBM, your computer won't let you use BASIC that way; instead, your computer will keep waiting for you to insert a disk into drive A.)

If the computer *did* find a disk in drive A or C (and is therefore supposed to say "A>" or "C>"), the computer typically asks for the date and time before printing the "A>" or "C>". But before printing the date and time, the computer checks whether the disk contains a file called "AUTOEXEC.BAT". If the computer finds such a file, it obeys the commands in that file, instead of printing the date and time.

For example, suppose you create an "AUTOEXEC.BAT" file that contains the command "dir". (To do that, type "copy con autoexec.bat"; underneath that, type "dir"; underneath that, press the F6 key and press the ENTER key.) The next time you turn on the computer, the computer will see that "AUTOEXEC.BAT" file, and will AUTOmatically EXECute it: the computer will print the directory instead of asking for the date and time.

Suppose you've been using the IBM PC for a while, and you hit some wrong keys, and the computer starts acting strangely, and you're so confused by the whole situation that you don't know what to do. When all else fails, you can turn the computer off, wait 10 seconds (for the RAM chips to cool down and forget whatever crazy stuff they were thinking of), and then turn the computer back on again. The computer will go through the entire boot-up procedure again: it will test the RAM, test the keyboard cable, decide which disk drive is the main one, and check whether that drive contains an "AUTOEXEC.BAT" file; if the computer finds that file, the computer will obey all the commands in it; but if the "AUTOEXEC.BAT" is missing, the computer simply asks for the date and time.

Instead of turning the computer off, waiting 10 seconds, and then turning the computer back on again, here's a short cut: hold down the Ctrl and Alt keys; and while you keep holding down those keys, tap the Del key. The computer will automatically go through the whole boot-up procedure again, except that it won't bother to test the RAM; it will still test the keyboard cable, find the main drive, and either obey "AUTOEXEC.BAT" or ask for the date and time.

Suppose you've written a BASIC program called FRED and saved it on the disk (by saying SAVE "FRED"). In the directory, the program will be listed as "FRED.BAS", because you called it FRED and it was written in BASIC. To make the computer run that program automatically when you turn on the computer, create an "AUTOEXEC.BAT" file that says "basic fred". To do that, just type this—

```
A>copy con autoexec.bat
basic fred
```

then press the F6 key and the ENTER key.

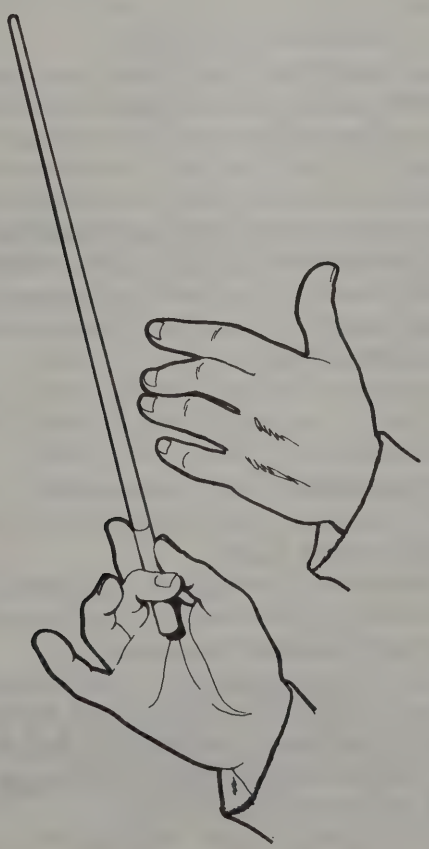
That's how to make the computer automatically run a BASIC program called FRED. For that procedure to work, the main disk must contain "AUTOEXEC.BAT" and "BASIC.COM" and "FRED.BAS".

In that example, FRED is called a *self-booting program*, because it runs automatically when you turn on the computer. To make a program self-booting, just create an "AUTOEXEC.BAT" file that tells the computer to run the program.

To make that "AUTOEXEC.BAT" file more professional, add a few extra lines, by typing this—

```
A>copy con autoexec.bat
echo off
cls
date
time
basic fred
```

and then pressing the F6 key and ENTER key. The "echo off" command stops the computer from printing excessive messages on the screen. The "cls" makes the computer clear the screen, so that the screen begins by being totally black. The "date" and "time" commands make the computer ask the human what the date and time are, so that the computer can put accurate dates and times into the directory. Finally, the "basic fred" makes the computer start doing BASIC and start running the program FRED.



COPYRIGHT PROBLEMS

For the computer to work properly, the disk in drive A must contain a file called "COMMAND.COM" and two hidden files ("IBMIO.COM" and "IBMDOS.COM"). Those three files are on the System Master and on every disk formatted by using "format/s b:".

To write a program, the normal procedure is to format a disk by using "format/s b:" and then put programs onto that disk. But if you're a professional programmer who puts fancy programs onto that disk, when you try to sell that disk to your clients you face a legal problem: those three files ("COMMAND.COM", "IBMIO.COM", and "IBMDOS.COM") are copyrighted by IBM, and it's illegal for you to sell anybody a disk containing those files, unless you pay IBM a royalty.

To avoid paying IBM a royalty, the typical programmer uses the following trick, which IBM approves of: the programmer formats a disk by saying "format/b b:", which is a special command that formats the disk, and reserves space on the disk for the three files special files, but doesn't put those special files onto the disk. When the client buys the disk from the programmer, the client puts "COMMAND.COM", "IBMIO.COM", and "IBMDOS.COM" onto that disk, by putting that disk into drive B, putting the System Master into drive A, and typing:

```
A>sys b:
A>copy command.com b:
```

The first line (sys b:) copies the system files "IBMIO.COM" and "IBMDOS.COM" to drive B. The second line (copy command.com b:) copies "COMMAND.COM". When the client has finished doing that copying, the disk in drive B is ready to use. The method is completely legal, since the client paid $65 to IBM for the System Master that contained those three files.

That's why, when you buy a disk containing software, the software's instruction manual typically says to put the disk into drive B, put your System Master into drive A, and type "sys b:" and "copy command.com b:".

Make sure you type "sys b:" first, and type "copy command.com b:" underneath. If you accidentally reverse the order, by typing "copy command.com b:" first, the computer will gripe.

SUBDIRECTORIES

You can create a *group* of files, and give that group a name.

For example, suppose you're planning to create a file called CHESS (that makes the computer play chess), and a file called PACMAN (that makes the computer play Pacman), and a file called BINGO (that makes the computer play bingo). You can turn CHESS, PACMAN, and BINGO into a group of files, called GAMES.

Suppose you're planning to write memos to your friends. You can make each memo be a file, and make that whole group of files be called MEMOS.

Suppose your friend ANN is sharing your disk. You can create, especially for her, a group called ANN. That group can contain just the files that interest her.

A group of files is called a *subdirectory*.

To create a subdirectory (group of files) called ANN, say "md ann", like this:

```
A>md ann
```

The "md ann" means "make a (sub)directory called ANN".

After you've said "md ann", you can put any file you wish into the ANN group, by putting "ann\" in front of the file's name.

For example, to create an ordinary file called SECRET (which is not in the ANN group), you'd say "copy con secret". To create a file called SECRET and put it into the ANN group, say "copy con ann\secret" instead.

To make LAMBCHOP be an ordinary copy of a file called MARY, you'd say "copy mary lambchop". To make LAMBCHOP be a copy of MARY and also make LAMBCHOP be part of the ANN group, say "copy mary ann\lambchop".

When you give commands about subdirectories (groups of files), make sure you type a backslash (\) and not a forward slash (/).

To see a directory of the ANN group, say "dir ann". The computer will print a directory showing all the files in the ANN group.

To manipulate a file in the ANN group, remember to put "ann\" before the file's name. For example, to erase SECRET from the ANN group, say "erase ann\secret". If LAMBCHOP's in the ANN group, and you want to see it on the screen, say "type ann\lambchop". To change the name LAMBCHOP to MEATBALL, say "rename ann\lambchop meatball"; the file will remain in the ANN group, but will be called MEATBALL instead of LAMBCHOP.

To erase all the files in ANN, say "erase ann*.*". The computer will ask, "Are you sure?" To reply, say "y".

To erase the entire group ANN, first erase all the files in ANN (by saying "erase ann*.*" and "y"); then say "rd ann", which removes the directory name ANN itself.

Short cut. If you're too lazy to type "ann\" repeatedly, say "cd ann", like this:

```
A>cd ann
```

It makes the computer automatically put "ann\" in front of every filename you mention.

For example, if you say "type lambchop", the computer will automatically do "type ann\lambchop" instead. If you say "copy con secret", the computer will do "copy con ann\secret" instead. If you say "dir lamb*.*", the computer will do "dir ann\lamb*.*" instead. If you say just "dir", the computer will do "dir ann" instead.

That all happens because you said "cd ann", which means "change the directory to ann".

When you've finished dealing with ANN and want to use other files instead, cancel the "cd ann", by saying "cd \".

Saying "cd ann" makes the computer use the subdirectory ANN. Saying "cd \" makes the computer use the regular directory instead, which is called the *root directory*.

Suppose you've invented a subdirectory called ANN and a subdirectory called SUE, and you've been using ANN, and you want to switch to SUE. One way is to first switch from ANN to the root directory, by saying—

```
A>cd \
```

and then switch from the root directory to SUE, by saying:

```
A>cd sue
```

You can combine those two commands into a single command, like this:

```
A>cd \sue
```

That's how to switch directly from ANN to SUE.

Should you use subdirectories? Although creating a subdirectory (such as ANN) helps you organize the files on your disk, subdirectories can confuse you: whenever you want to use a file, you must remember which subdirectory it's in; and to use the files in ANN, you must remember to say "ann\" or "cd ann". Another disadvantage of subdirectories is that the computer handles a subdirectory's files more slowly than it handles normal files.

If you're using floppy disks, subdirectories aren't practical. Instead of creating a subdirectory called ANN, just give your friend Ann her own floppy disk!

Use subdirectories only if you're using a hard disk, which must be shared by several people or activities.

WOW! YOU FINISHED CHAPTER 6: DOS!
TURN THE PAGE FOR CHAPTER 7: WORD PROCESSING.

BACKGROUND

WHAT'S WORD PROCESSING?

A *word-processing program* helps you write and edit sentences and paragraphs of information. For example, it can help you write and edit a business letter, a magazine article, or even a whole book. Whatever you're writing and editing (a business letter, magazine article, or whole book) is called the *document*.

Use a word-processing program only if the document consists of sentences and paragraphs. If the document consists mainly of pretty drawings, use a *graphics program* instead. If the document is a table of numbers, use a *spreadsheet program* instead. If the document is a list of names (such as a list of your customers), use a *data-management program* instead. If the document is an outline (composed of topics and subtopics), use an *outline processor* instead.

To use a word-processing program, you put your fingers on the keyboard, type the sentences and paragraphs that make up your document, and then edit that document, by using special keys on the keyboard. What you type and edit appears on the screen. When you've edited what's on the screen, you can make the computer copy the document from the screen to the disk, or from the screen to the printer, so that the document appears on paper.

HOW THE TERM "WORD PROCESSING" WAS INVENTED

When computers were first invented, they were used mainly to handle lists of numbers, names, and addresses. Using computers for those purposes was called *data-processing (DP)*, and computer centers were called *data-processing centers (DP centers)*, which were run by a group of programmers and administrators called the *data-processing department (DP department)*.

Unfortunately, most of those old computer systems were complicated, expensive, needed repairs every day, gave wrong answers because they were programmed incorrectly, needed continuous reprogramming and supervision by the programmers and administrators, and turned into bureaucratic and technological nightmares. So the term "data-processing" got a bad reputation.

Secretaries who wanted to write and edit reports preferred to use simple typewriters, rather than deal with the dreaded "data-processing department".

When easy-to-use word-processing programs were finally invented for computers, secretaries were afraid to try them, because computers had developed a scary reputation. The last thing a secretary wanted was a desktop computer, which the secretary figured would mean "desktop trouble".

That's why the term "word-processing" was invented. Manufacturers said to the secretaries, "We know you don't like computers and data-processing equipment. But don't worry: the machines we want to put on your desks are *not* computers; they're just souped-up typewriters. You like typewriters, right? Then you'll like these cute little machines also. We call them *word processors*. Don't worry: they're not data-processing equipment; they're not computers."

The manufacturers were lying: their desktop machines *were* computers. But the manufacturers tried to hide the fact that they were computers, by calling them *word processors* instead, and by not including any software that dealt with numbers or lists.

The trick worked: secretaries bought the word processors, especially the *Wang Word Processor* and the *IBM Displaywriter*.

Today, secretaries are less afraid of computers. Most secretaries buy the IBM PC, which they realize is a computer, and then buy a word-processing program for it.

THE THREE DEFINITIONS OF "WORD PROCESSOR"

Strictly speaking, a "word processor" means "a computer (usually desktop) whose main purpose is to do word processing". But the term "word processor" is also used loosely, to mean "a word-processing program" or "a typist who uses a word-processing machine or a word-processing program".

In advertisements, a "$5000 word processor" is a machine; a "$200 word processor" is a program that you put into a computer; a "$9/hour word processor" is a typist who knows about word processing.

THE TWO BEST WORD-PROCESSING PROGRAMS

Of all the word-processing programs on the market today, the two I recommend the most highly are *Writing Assistant* and *Word Perfect*. They both run on the IBM PC. (A stripped-down version of Word Perfect runs on the Apple 2e & 2c.)

Of all the decent word-processing programs, Writing Assistant is the easiest to learn how to use. Unfortunately, it cannot handle advanced features, such as footnotes.

If you need a word-processing program that can handle advanced features, get Word Perfect. Of all the decent word-processing programs, Word Perfect is the fanciest. But it's harder to learn how to use than Writing Assistant and also costs more.

Since Writing Assistant is the easiest and costs very little, I recommend it to beginners and to people who use word processing lightly (just a few times a month). Since Word Perfect is the fanciest, I recommend it to professional writers and experienced secretaries who need advanced word processing daily.

Although Writing Assistant and Word Perfect are what I recommend to *most* people, they might not be the best for *you*, because some other word-processing program might more closely match your own personal needs, desires, habits, and style.

Let's begin by examining Writing Assistant and the IBM version of Word Perfect. Afterwards, we'll examine all the other popular word-processing programs for microcomputers.

Publisher. Writing Assistant is published by IBM itself. Word Perfect is published by a strange company in Utah, called Satellite Software International (288 W. Center St., Orem, UT 84057, phone 801-224-4000).

Price. Word Perfect is expensive: it lists for $495, though you can get it for just $197 from mail-order discount dealers. Writing Assistant costs less: it lists for just $149, and IBM sometimes gives it away free when you buy an IBM computer.

Version numbers. The current version of Word Perfect is excellent. It's called "version 4.1", and uses PC-DOS. It's an improvement on the previous version, which was called "version 4.0".

The original version of Writing Assistant was called "version 1.00"; it used PC-DOS. Then IBM developed a fancier version, called "version 1.01", which could handle *two* operating systems: PC-DOS and Topview. Unfortunately, version 1.01 makes a mistake when you try to delete part of what you typed; I've notified IBM of the mistake, and I hope IBM will develop a version 1.02 that corrects it. But since IBM never admits to mistakes and never announces future products, we'll have to wait and see. In the meantime, avoid version 1.01 and use the old version 1.00 instead, which didn't make any mistakes.

Related products. Word Perfect was originally written for a Data General minicomputer, and later rewritten for the IBM PC. Besides Word Perfect 4.1, you can also buy stripped-down versions, such as Personal Word Perfect ($195 list), Word Perfect Junior ($89 list, $39 at discount), and the Apple 2e & 2c version ($179 list, $77 at discount). Versions for the IBM PC are available in French, German, Spanish, Dutch, Danish, Swedish, Norwegian, and Finnish.

Writing Assistant is IBM's improvement on *PFS Write Version A*, an older word-processing program developed by Software Publishing Company. Recently, Software Publishing Company has developed *PFS Write Version C*, which is fancier than Writing Assistant but harder to learn.

Memory requirements. Word Perfect requires at least 256K of RAM and two disk drives. Writing Assistant, which isn't as fancy, needs just 128K of RAM and one disk drive.

Phone support. If you bought Word Perfect and have any difficulty using it, you can get free technical help by phoning the publisher's customer-support hot-line at 801-227-4040 (out-of-state 800-321-5906).

If you have a technical question about using Writing Assistant, too bad: IBM doesn't provide any hot-line. Instead, you must ask the salesperson who sold you the program (and who probably doesn't know much). On the other hand, since Writing Assistant is so easy to use, and since its manual is so easy to read, you probably won't have questions anyway.

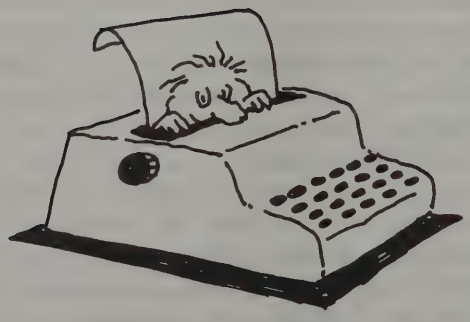
Supplementary disks. Writing Assistant comes on a single disk, called the Program Disk. IBM also gives you a Sampler Disk, which includes a few cute examples of documents.

Word Perfect 4.1 comes on several disks: the main disk, the Speller Disk (which corrects your spelling), the Thesaurus Disk (which includes a thesaurus), the Printer Disk (which customizes the program to handle different printers), and the Learning Disk (which contains cute examples).

Copying the program. Before using the program the first time, make a backup copy of it.

To copy the Word Perfect disks, put the PC-DOS System Master Disk into drive A, turn on the computer, type the date and time, say "format/s b:", put a blank disk into drive B, press the ENTER key, put the Word Perfect disk you want to copy into drive A, and say "copy *.* b:". Word Perfect lets you make as many copies as you like.

Writing Assistant is nasty: it refuses to make more than one backup copy. To make the one copy that Writing Assistant permits, put the PC-DOS System Master Disk into drive A, turn on the computer, type the date and time, remove the write-protect tab from the Writing Assistant Sampler Disk, put that disk into drive A, type the word "backup" (followed by the ENTER key), remove the write-protect tab from the Writing Assistant Program Disk, put that disk into drive A, put a blank disk into drive B, press the ENTER key again, and follow the instructions on the screen.



QUICK TRICKS

GETTING TO PAGE 1

To start using Writing Assistant, put the PC-DOS System Master Disk into drive A, turn on the computer, type the date and time, put the Writing Assistant Program Disk into drive A, and after the "A>" type the word "write". The computer will print this on the screen:

```
Main Menu

1. Type/Edit
2. Define page
3. Print
4. Get/Save/Remove
5. Clear

9. Exit
```

That main menu is a list of the various activities the program can perform for you. To begin using Writing Assistant, choose activity 1 from the menu (Type/Edit). To do that, tap the 1 key and then the ENTER key. The lower right-hand corner of the screen will say "Line 1 of Page 1", which means you can begin typing your document.

To start using Word Perfect, put into drive A your *copy* of Word Perfect's main disk, which you made by using "format/s b:" and "copy *.* b:". Do *not* insert the original! Put a blank formatted disk into drive B. Turn on the computer. Type the date and time. Wait for the "A>", type "b:", wait for the "B>", and type "a:wp", so that your screen looks like this:

```
A>b:
B>a:wp
```

Then a message will appear on your screen. Typically, the message says "Doc 1 Pg 1 Ln 1", which means you can begin typing your document. But if you haven't used Word Perfect before, you'll see a message saying "Welcome to Word Perfect"; press the ENTER key; you'll see another introductory message; press the ENTER key again; and finally you'll see "Doc 1 Pg 1 Ln 1", which means you can begin typing your document.

TYPING THE DOCUMENT

Begin typing whatever document you wish to create. For example, try typing a novel that begins like this:

Once upon a time, a man was walking down the street, when lo and behold, his house was gone. As he gaped into the hole, a burning sensation in his shoes warned him that . . .

I'll let you complete that paragraph yourself! Be creative!

The SHIFT keys, CAPS LOCK key, and BACKSPACE key (which is next to the "=" key) all work the way you expect.

As you type that paragraph and get near the right margin, do *not* press the ENTER key: just keep on typing! The computer will press the ENTER key for you automatically. If you try to type a long word near the right margin, and the word is too long to fit before the margin, the computer will automatically move the entire word to the line below. The computer's ability to automatically move an entire word to the line below is called *word wrap*. Since the computer automatically presses the ENTER key for you, *never press the ENTER key yourself*, until you reach the end of a paragraph.

Press the ENTER key only at the end of the paragraph. Pressing the ENTER key there makes the computer return to the left margin, so that you can begin a new paragraph.

If you want to double-space between paragraphs, press the ENTER key *twice*. If you want to indent the new paragraph, press the TAB key, which is next to the Q key and has arrows on it.

If you're using an obsolete version of Word Perfect, the computer will occasionally say "Position hyphen"; to reply, tap the F1 key. (Version 4.1 doesn't have that hassle.)

MOVING THE CURSOR

After you've typed a few paragraphs (and pressed the ENTER key at the end of each paragraph), you can move around the screen and edit what you've typed.

On your screen, the short blinking underline is called the *cursor*. To move the cursor up, press the key that has an up-arrow on it. You can move the cursor in all four directions, by pressing the up-arrow, down-arrow, left-arrow, and right-arrow keys. Each of those keys automatically repeats: so to move the cursor up *several* lines, just keep your finger on the up-arrow key a while.

If you tap the left-arrow key while holding down the CONTROL key (which says "Ctrl" on it), the cursor will hop to the previous word. If you tap the right-arrow key while holding down the CONTROL key, the cursor will hop to the *next* word.

To move the cursor farther, Writing Assistant uses these keys:

Keys you press	Where the cursor will move
HOME	the beginning of the line
END	the end of the line
Ctrl with HOME	the beginning of the document
Ctrl with END	the end of the document
PAGE UP (PgUp)	the previous screenful
PAGE DOWN (PgDn)	the next screenful

Word Perfect uses these keys instead, which are harder to learn but offer you more choices:

Keys you press	Where the cursor will move
HOME then ←	the beginning of the line
HOME then →	the end of the line
HOME then ↑	top of screen; if there already, top of previous screen
HOME then ↓	bottom of screen; if there already, bottom of next screen
HOME HOME then ↑	the beginning of the document
HOME HOME then ↓	the end of the document
PAGE UP (PgUp)	the previous "page"
PAGE DOWN (PgDn)	the next "page"
Ctrl HOME 3 ENTER	the 3rd "page"

In Word Perfect, the screen displays 24 lines of your document, whereas a "page" typically consists of 54 lines, because that's how many lines can fit on a sheet of paper (allowing room for margins).

Word Perfect permits some short-cuts. To reach the end of the line, you can simply tap the END key, instead of pressing "HOME then →". Tapping the "-" key that's next to the PgUp key is a short-cut for "HOME then ↑". Tapping the "+" key that's next to the PgDn key is a short-cut for "HOME then ↓".

DELETING A CHARACTER

To delete the character you just typed, press the BACKSPACE key. To delete a character you typed long ago, move the cursor to that character, then press the DELETE key (which says "Del" on it).

REPLACING

If your document contains incorrect characters and you want to replace them, move the cursor to where the incorrect characters begin.

If you're using Writing Assistant, make sure the bottom of your screen says the word "Replacing"; if you're using Word Perfect, make sure the bottom of your screen says the word "Typeover". (If you *don't* see such a word at the bottom of your screen, tap the "Ins" key.)

Then retype the characters correctly.

INSERTING

To insert extra characters into the middle of your document, move the cursor to where you want the extra characters to begin.

If you're using Writing Assistant, make sure the bottom of your screen says the word "Inserting"; if you're using Word Perfect, make sure the bottom of your screen does *not* say "Typeover". (If the bottom of your screen isn't correct, tap the "Ins" key.)

Then type the characters you want to insert. The other characters on the screen will automatically move out of the way, to make room for the extra characters.

If you're using Word Perfect, inserting characters will make other characters move off the right edge of the screen and temporarily disappear; but they'll come back onto the screen again when you move the cursor. (If you're using an obsolete version of Word Perfect, the characters won't come back onto the screen until you move the cursor *down*.)

USE YOUR TEMPLATE

WHAT THE TEMPLATE MEANS

At the left side of the IBM PC keyboard, you'll see ten *function keys*, labeled F1, F2, F3, F4, F5, F6, F7, F8, F9, and F10.

When you buy a sophisticated program (such as Writing Assistant and Word Perfect), you get with it a piece of stiff paper that fits around the function keys. The paper's called a *template*. On the paper, you'll see notes telling you the purpose of each function key.

For example, when you buy Writing Assistant, the paper template shows this note next to the F6 key:

```
ERASE LINE
ERASE WORD
```

That note means: if you press the F6 key, you'll ERASE a WORD; if you press the F6 key while holding down the SHIFT key, you'll ERASE a LINE.

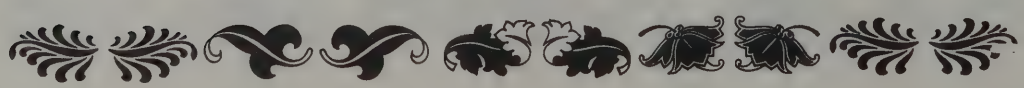
Writing Assistant makes the function keys perform 15 functions:

Function key	What the computer will do
F1	HELP you, by reminding you how to use Writing Assistant
F2	CHECK your SPELLING
F3	ADJUST the line, by centering it or by shifting it over
F3 with SHIFT	CHANGE the COLOR of the screen
F4	SET the TABS
F5	EMPHASIZE, by using boldface or underlining or colored ink
F5 with SHIFT	APPEND another document to this one
F6	ERASE the WORD where the cursor is
F6 with SHIFT	ERASE a LINE
F7	MOVE a BLOCK of text, to a different place in the document
F7 with SHIFT	DELETE a BLOCK of text from the document
F8	COPY a BLOCK of text, to a different place in the document
F8 with SHIFT	REUSE a BLOCK of text that you had deleted
F9	SEARCH for a phrase AND REPLACE it by a different phrase
F10	CONTINUE a process that was begun but not completed

Word Perfect is fancier; it makes the function keys perform 40 functions:

	The key itself	The key with SHIFT	The key with ALT	The key with CTRL
F1:	CANCEL	SUPER/SUBSCRIPT	use THESAURUS	outer SHELL
F2:	SEARCH RIGHT	SEARCH to LEFT	REPLACE a phrase	check the SPELLing
F3:	HELP explain	SWITCH documents	REVEAL the CODES	change the SCREEN
F4:	INDENT RIGHT	INDENT CENTERED	define a BLOCK	MOVE some text
F5:	LIST all FILES	print the DATE	MARK the TEXT	send TEXT IN/OUT
F6:	do BOLDface	CENTER the line	FLUSH to RIGHT	TAB ALIGN
F7:	EXIT program	PRINT on paper	MATH/COLUMNS	insert FOOTNOTE
F8:	UNDERLINE	FORMAT the LINE	FORMAT the PAGE	FORMAT the PRINT
F9:	MERGE RETURN	MERGE END record	put MERGE CODES	MERGE or SORT
F10:	SAVE onto disk	RETRIEVE from disk	use a MACRO	MACRO DEFINE

Let's try some practical examples. . . .



UNDERLINING

Suppose you're typing a document and want to underline the next phrase. Here's how.

Word Perfect. Say UNDERLINE, by pressing the F8 key. Then type the phrase you want to underline. Then say UNDERLINE again (which ends the underlining).

Writing Assistant. Type the phrase you want to underline, then move the cursor back to the beginning of the phrase. Say EMPHASIZE, by pressing the F5 key. Hold down the U key a while (which stands for "Underline"), until all the characters in the phrase have been emphasized. Then say CONTINUE, by pressing the F10 key.

Using a color monitor? If you're using a color monitor, the characters you say to underline will change colors on the screen, but won't be actually underlined until you print them on paper.

BOLDFACING

Here's how to boldface the next phrase, so that it becomes extra dark when it's printed on paper.

Word Perfect. Say BOLD (by pressing F6), then type the phrase, then say BOLD again (which ends the bolding).

Writing Assistant. Type the phrase, then move the cursor back to the beginning of the phrase. Say EMPHASIZE (by pressing F5). Hold down the H key a while (which stands for "Highlight"), until the entire phrase has been highlighted. Then say CONTINUE (by pressing F10).

CENTERING

Suppose you've finished typing a paragraph (and pressed the ENTER key afterwards). Here's how to make the next line to be a centered title.

Word Perfect. Say CENTER (by pressing SHIFT with F6), then type the title, then press the ENTER key.

Writing Assistant. Type the title (without pressing the ENTER key). Say ADJUST (by pressing F3), then tap the C key (which stands for "Center"). Then tap the down-arrow key.

BLOCKS

You can manipulate a large portion of your document with a single keystroke! The portion you're manipulating is called the *block*. It can consists of several words, or several sentences, or several paragraphs, or even several pages.

To manipulate a block, first move the cursor to the beginning of the block. (For example, to manipulate a whole paragraph, move the cursor to the beginning of the paragraph.) What you do next depends on which word-processing program you use.

Writing Assistant. If you're using Writing Assistant, you can either move the block or delete it.

To move the entire block to a different place in your document, say MOVE BLOCK (by pressing the F7 key). The computer will tell you, "Put the cursor at the end of the block, then press the F10 key". Do so. The computer will say, "Put the cursor where you want the block to appear, then press the F10 key." Do so.

To delete the block instead of moving it, say DELETE BLOCK instead of MOVE BLOCK. The computer will tell you, "Put the cursor at the end of the block, then press the F10 key". Do so. The computer will ask, "Are you sure you want to delete that entire block (Y/N)?" Tap the Y key, to confirm that yes, you really want to delete that entire block.

Writing Assistant can't move long blocks. The longest block it can move is about 50 lines. If you want to move a chunk that's much more than 50 lines, you must divide the chunk into small blocks, and move each block separately, one at a time.

If you're using version 1.00 of Writing Assistant, the block commands work properly (for short blocks). If you're using version 1.01 instead, which is defective, the block commands accidentally leave a large gap near the right margin; to eliminate the gap, move the cursor to "almost the beginning of the gap" (move it to the second space in the gap) and then press the DELETE key.

Word Perfect. If you're using Word Perfect, say BLOCK (by pressing the ALT key with the F4 key), then move the cursor to the end of the block (which can be as long as you wish!), then say what you want to do to the block.

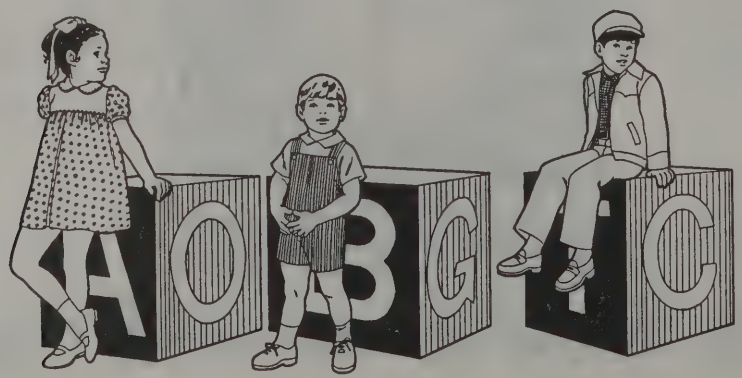
If you want the entire block to be underlined, say UNDERLINE (by pressing the F8 key).

If you want the entire block to be boldfaced, say BOLD (by pressing the F6 key).

If you want the entire block to be deleted, tap the DELETE key (which says "Del"). The computer will ask, "Delete Block (Y/N)?". Tap the Y key, to confirm that yes, you really want to delete the whole block.

If you want to move the entire block to a different place in your document, say MOVE (by pressing the CONTROL key with the F4 key). Then tap the 1 key (which means "Remove the block from its current location"), then put the cursor where you want the block to appear, then say MOVE again, then tap the 5 key (which means "Put the block here").

After you've deleted or moved a block, you'll see a large gap near the right margin. Don't worry: the gap will eventually go away. (If you're eager to see it go away immediately, move the cursor to the next paragraph.)



FINAL STEPS

YOUR LAST THREE STEPS

After you've edited your document, you'll want to do three things:

Copy the document onto paper.
Copy the document onto a disk.
Move on to a different document or task.

Here's how. . . .

WRITING ASSISTANT'S WRAP-UP

When you've finished typing and editing a document by using Writing Assistant, press the ESCAPE key (which says "Esc"). The computer will display the main menu again:

```
Main Menu

1. Type/Edit
2. Define page
3. Print
4. Get/Save/Remove
5. Clear

9. Exit
```

To print on paper, choose "Print" from the main menu, by typing a 3 and then pressing the ENTER key. To confirm that you really mean it, press the ENTER key again, then pause, then press the ENTER key *again*. The computer will print the entire document onto paper. Then the computer will display the main menu again, so you can make another selection.

To manipulate the disks, choose "Get/Save/Remove" from the main menu, by typing a 4 and then pressing the ENTER key. The computer will display this sub-menu:

```
Get/Save/Remove Menu

1. Get document
2. Save document
3. Remove file
```

To copy your document onto disk B and call it FRED, tap the 2 key (which stands for "Save document"), then tap the TAB key, then type:

```
b:fred
```

Then press the ENTER key. The computer will save your document onto disk B and call it FRED. Then the computer will display the main menu again, so you can make another selection.

To erase the screen, so you can begin typing a new document, choose "Clear" from the main menu, by typing a 5 and then pressing the ENTER key. To confirm that you really mean it, press the ENTER key again. The computer will clear the screen and let you begin typing a new document. When you've finished typing the document, press the ESCAPE key, to make the computer display the main menu again.

To retrieve an old document that you had saved on disk B, choose "Get/Save/Remove" from the main menu, by typing a 4 and then pressing the ENTER key. The computer will display the Get/Save/Remove submenu. To retrieve FRED from disk B, tap the 1 key (which stands for "Get document"), then tap the TAB key, then type:

```
b:fred
```

Then press the ENTER key. To confirm that you really mean it, press the ENTER key again. The computer will get FRED from disk B and display that document on your screen, so that you can see it and edit it. When you've finished editing it, press the ESCAPE key, to make the computer display the main menu again.

To see a directory of all the files on disk B, choose "Get/Save/Remove" from the main menu, then tap the 1 key (which stands for "Get"), then tap the TAB key, then type:

```
b:
```

Then press the ENTER key. The computer will display a directory of disk B. To use one of those documents, type the document's name, then press the ENTER key; but if you *don't* want to use any of those documents yet, press the ESCAPE key instead.

If you ever want to erase a document from disk B, choose "Get/Save/Remove" from the main menu, then tap the 3 key (which stands for "Remove"), then tap the TAB key, then type the name of the document you want to erase from the disk (remember to put a "b:" at the beginning of the name!), then press the ENTER key, then press the ENTER key again (to confirm). The computer will remove the document from the disk and then return you to the main menu.

Whenever you've finished using Writing Assistant, choose "Exit" from the main menu, by typing a 9 and then pressing the ENTER key. Then press the ENTER key again, to confirm. The computer will say:

```
A>
```

WORD PERFECT'S WRAP-UP

To make Word Perfect print your document onto paper, say PRINT (by pressing SHIFT with F7). The computer will say:

```
1 full text    2 page    3 change options    4 printer control    5 type-thru
```

To print the entire document, tap the 1 key. To print just the page that the cursor's on, tap the 2 key. (Tap 3, 4, or 5 only if you want an abnormal printing method.)

When you've finished editing and printing your document, say EXIT (by pressing F7). The computer will ask, "Save document? (Y/N)". Tap the Y key, to signify that yes, you want to save the document. The computer will say, "Document to be saved." Type a name for your document, then press the ENTER key. The computer will copy the document onto disk B.

Then the computer will ask, "Exit WP? (Y/N)". If you're *not* done using Word Perfect, tap the N key: the computer will erase the screen, so that you can begin typing a new document. If you *are* done using Word Perfect, tap the Y key instead: the computer will stop using Word Perfect and will say "B>".

After the computer says "B>", return your computer back to its normal state, by saying "a:", so your screen looks like this:

```
B>a:
```

Then press the ENTER key. The computer will say:

```
A>
```

Tricks. If you mess up a document so badly that you want to erase it all and start over, say EXIT (by pressing F7). Then tell the computer not to save the mess: tap the N key. Then tell the computer not to exit from Word Perfect: tap the N key again. The computer will erase the entire screen and let you start over.

If you want to save a document without clearing the screen, say SAVE (by pressing F10), then type a name for your document, then press the ENTER key.

Suppose you're typing a document onto your screen, and you want to put in the middle of it the complete text of FRED, which is on disk B. To do that, move the cursor to the place on the screen where you want to insert FRED, then say RETRIEVE (by pressing SHIFT with F10), then say:

```
fred
```

Then press the ENTER key. The computer will copy FRED from disk B to the middle of your screen, where the cursor is. So the document on your screen is now longer: it includes everything that was on the screen before, plus the complete text of FRED.

To see a directory of disk B, say LIST FILES (by pressing F5), then press the ENTER key. The screen will display a directory of all the files on disk B. Move the cursor to the name of whichever file interests you. That file's called the *target file*. At the bottom of the screen you'll see this menu:

```
1 retrieve    2 delete    3 rename    4 print    5 text in
6 look    7 change directory    8 copy    9 word search
```

If you tap the 1 key (which means "retrieve"), the computer will retrieve the target file and insert that entire file into the middle of the document you've been working on. If you tap the 2 key ("delete") and then tap the Y key (to confirm that yes, you really want to delete the entire file), the computer will erase the target file from the disk. If you tap the 3 key ("rename"), then invent a new name (followed by the ENTER key), the computer will change the target file's name to the new name you invented. If you tap the 4 key ("print"), the computer will print the entire target file onto paper. Tap 5 ("text in") to retrieve a file created by a word-processing program *other* than Word Perfect. Tap 6 ("look") to peek at what the target file contains; when you've finished peeking, tap the ENTER key. (Tap 7, 8, or 9 only if you want abnormal disk manipulation.) If you tap the ENTER key instead of a number from 1 to 9, the directory will disappear from the screen and you'll return to the document you've been working on.

ADVANCED FEATURES

CHANGING THE MARGINS

All the popular word-processing programs let you change the left and right margins. Most word-processing programs (such as Writing Assistant and Word Perfect) let you change the top and bottom margins also.

If you're a student who wrote an eight-page paper, and your teacher requires it to be ten pages long instead, just make the margins wider and—presto!—you suddenly have a ten-page paper.

Word Perfect permits multiple margins, so that you can make some paragraphs in your document narrower than the others. Writing Assistant does *not* permit multiple margins: it insists that all paragraphs have the same width as each other.

RIGHT JUSTIFICATION

Most word-processing programs let you choose between two styles of right margin.

The fancy style is called *right-justified*: the computer makes the right margin perfectly straight, by inserting extra spaces between the words in the line. The simpler style is called *ragged-right*: the right margin is somewhat uneven, as if it were produced by an ordinary typewriter.

Although the fancy style looks more professional (and is therefore used by professional publishers of newspapers, magazines, and books), the simpler style looks more friendly, and is therefore more appropriate for personal correspondence.

Try this experiment: look at ads in newspapers, and notice which ads are right-justified (to give a "professional" image) and which ads are ragged-right (to give a more "friendly" image).

Unfortunately, Writing Assistant can't right-justify: it produces only ragged-right. So Writing Assistant doesn't appeal to snobs who want to look professional. On the other hand, Writing Assistant is wonderful for cheery personal correspondence. Word Perfect, which is fancier, can handle both styles.

HEADERS & FOOTERS

Most word-processing programs let you print page numbers at the top or bottom of each page. Most let you print additional comments next to the page numbers. Comments next to page numbers at the top of each page are called *headers*; comments at the bottom of each page are called *footers*.

If you're using Writing Assistant, the headers and footers that you create are printed automatically on every page and are centered—whether you like it or not! Word Perfect is more flexible: it lets you create *two* kind of headers and footers—one set for the odd-numbered pages, and the other set for the even-numbered pages—and for each kind lets you decide whether to center it or not.

TABS

When you press the TAB key, the cursor moves to the right until it reaches the next *tab stop*. All the popular word-processing programs let you change the positions of the tab stops.

When you type a column of numbers, you want their decimal points to line up, like this:

```
        74.9
5,382,931.726
        -.82
```

All the popular word-processing programs help you line up those decimal points, by using a trick called *decimal tabs*. Here's how. . . .

First, create a tab stop where you want the decimal points to be.

To type a number (such as 74.9), tab to that tab stop (using a special decimal-tab method), then type the entire number (74.9). As you type the number, the computer will automatically shift it to the left the correct distance, so that the decimal point winds up at the tab stop.

SEARCH & REPLACE

All the popular word-processing programs let you search for a phrase and replace it by a different phrase.

For example, if you write a letter that talks about Fred, and then want to write a similar letter about Sue, you can tell the computer to scan through your document and change each "Fred" to "Sue". If you write a book about "How to be a better salesman", and then a feminist tells you to change each "salesman" to "salesperson", you can make the computer change them all automatically. If you're writing a long advertisement that mentions "Calvin Klein's Hot New Flaming Pink Day-Glo Pajamas" repeatedly, and you're too lazy to type that long phrase so often, just type the abbreviation "CKP"; when you've finished typing the document, tell the computer to replace each "CKP" by the long phrase it stands for.

If you tell the computer to change each "my" to "our", will the computer change "myth" to "ourth"? If you tell the computer to change each "my" to "our", will the computer also change each "My" to "Our", and change each "MY" to "OUR"?

Writing Assistant handles all those problems well: if you say to change each "my" to "our", it will *not* change "myth" to "ourth", but *will* change each "My" to "Our" and change each "MY" to "OUR". Word Perfect, unfortunately, isn't as bright. (That's one of the few ways in which Word Perfect is inferior.)

SPELLING

Writing Assistant comes on a disk that also includes a list of 125,000 popular words. That list is called a *dictionary*. Although that "dictionary" does *not* include definitions of those words, it *does* include the correct spelling for each of them. If you move the cursor to the beginning of your document and then say CHECK SPELLING (by pressing the F2 key), the computer will check each word in your document against the dictionary, to make sure all the words in your document are spelled correctly. If the computer finds a word in your document that's not in the dictionary, the computer will warn you and also try to guess what you really meant! The computer will tell you the six words in the dictionary that most closely resemble the word you typed. For example, suppose you didn't know how to spell "restaurant" and typed "restront" instead. When the computer sees "restront", the computer will warn you that "restront" is not in the dictionary and will suggest these six alternatives: "restraint", "restoring", "restaurant", "restrain", "restraints", and "restrict".

Word Perfect includes a dictionary also.

Although Writing Assistant and Word Perfect include dictionaries for free, some other word-processing programs provide dictionaries only as extra-cost options. When selecting a word-processing program, ask whether the dictionary costs extra.

When you've obtained a dictionary, you can add your own words to it. But when typing your new words into the dictionary, make sure you spell them correctly!

Even if you know how to spell, make the computer check your document against its dictionary, because the computer will find and correct your accidental typographical errors. The computer can spot minor typographical errors that human proofreaders overlook.

After the computer checks your document, make a human check your document also, because only a human can notice that the sentence "I live you" is wrong, and should be "I love you" instead.

ADDENDA

Word Perfect includes commands that help you easily create footnotes (automatically numbered at the bottom of each page), endnotes (which resemble footnotes but appear at the end of the document instead of at the bottom of each page), a table of contents (showing all your titles and page numbers), an index (showing all the document's important words in alphabetical order, with their page numbers), an outline (showing the subtopics indented under the topics), and lists (of diagrams, charts, and tables).

Moreover, Word Perfect is very flexible: for each of those addenda, you can select your own style. For example, in the index, you can make the computer put the page numbers in parentheses, or without parentheses and next to the right margin, or however else you wish.

Writing Assistant has none of those automatic features: if you're using Writing Assistant, you must create all those addenda manually.

MACROS

After you've been using a word-processing program for a while, you'll find yourself using certain sequences of commands and keystrokes often. Instead of forcing you to type such a sequence repeatedly, Word Perfect lets you invent a short name for the sequence; whenever you type that name again, the computer will automatically perform the whole sequence of commands and keystrokes that the name stands for.

The name you invent is called a *macro*. Most fancy word-processing programs let you invent macros.

Writing Assistant, which isn't so fancy, doesn't permit macros.

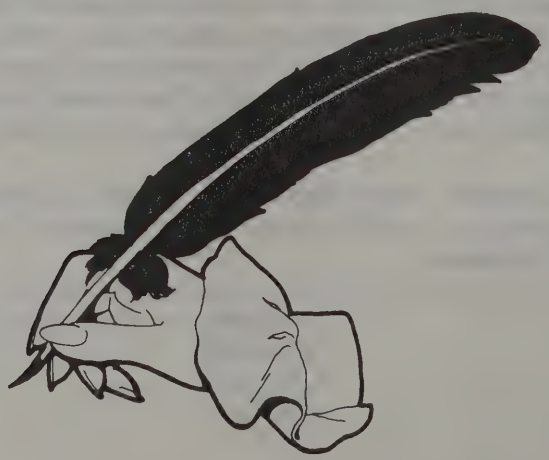
PRINTER DRIVERS

Some printers can perform fancy tricks. They can print characters that are underlined, boldfaced, or in italics. They can make the characters different sizes: extra-wide, or extra-narrow, or extra-short, or extra-tall. They can print characters that have European accents on them. They can print Japanese characters. They can print graphics.

Unfortunately, the printer industry isn't standardized: each printer has its own special tricks, and its own special codes for making those tricks occur. The part of the word-processing program that handles the printer's peculiarities is called the *printer driver*. Before you buy a word-processing program and a printer, make sure the word-processing program includes a printer driver that's designed specifically for the printer you own.

The current version of Writing Assistant (version 1.01) comes with 9 printer drivers, for 9 popular kinds of printers: Okidata, NEC, IDS, C. Itoh, old Epson, Epson with Graftrax Plus, IBM Graphics, IBM Color, and plain.

Word Perfect, which is fancier, comes with 31 printer drivers. In addition, Word Perfect's publisher has developed 32 other printer drivers, available upon request. So altogether, Word Perfect can handle 63 kinds of printers without requiring any effort from you. Moreover, Word Perfect comes with a program that helps you invent your *own* printer driver easily, so that you can make Word Perfect handle *any* kind of printer you wish!



WHAT TO BUY

WORD PROCESSORS FOR THE IBM PC

A word-processing program should be easy, should perform fancy tricks, should run quickly, and should cost little. Using those criteria, let's rate the most famous word-processing programs for the IBM PC:

Word processor	Easy?	Fancy?	Quick?	Cheap?	Weighted avg.	Why popular
Word Perfect 4.1	B	A+	A	C	2.76 = A-	Fancy
Writing Assistant	A+	D	B	B	2.24 = B+	Easy
PFS Write C	A	C	B	A	2.18 = B+	Easy
Samna Word 3	B	A	D	C	2.06 = B	Fancy
Leading Edge WP	B	B	B	A	2.06 = B	Moderate
Microsoft Word 2	C	A+	C	C	2.06 = B	Fancy
Xywrite 3	C	A	A+	C	2.06 = B	Fancy
Wordstar 2000 Plus	B	A	D	D	2 = B	Fancy
Volkswriter 3	B	B	B	B	2 = B	Moderate
Einstein Writer	B	B	B	B	2 = B	Moderate
Textra 3.1A	B	C	A+	A+	2 = B	Cheap
Office Writer 4	B	B	B	C	1.94 = B	Historic
PC-Write	C	B	A+	A+	1.89 = B	Cheap
Easy	B	C	D	A	1.47 = C+	Historic
Wordstar Profess'l	C	B	B	C	1.47 = C+	Historic
Bank Street Writer	B	D	B	A	1.35 = C+	Historic
Homeword	B	D	B	A	1.35 = C+	Historic
Multimate Profess'l	C	B	D	C	1.24 = C+	Historic
Multimate Advantage	C	B	D	D	1.18 = C+	Historic
Displaywrite 3	D	B	B	D	.94 = C	Historic
Just Write	C	D	D	A	.65 = C-	Historic

In that chart, D means poor (0 points), C means fair (1 point), B means good (2 points), A means excellent (3 points), and A+ means absolutely amazing (4 points).

To get an A+ for being easy, the program must come with an easy-to-read manual, have commands that are easy to remember, continually print messages that remind you what to do next, print an easy-to-read error message whenever you make a mistake, and be so hassle-free that using it feels "natural".

To get an A+ for being fancy, the program must be able to keep the right margin perfectly straight and include commands for handling advanced problems such as spelling, footnotes, the table of contents, and the index.

To get an A+ for being quick, the program must accomplish each popular command instantaneously, without making you wait.

To get an A+ for being cheap, the program must be available legally for $25 or less (plus shipping costs and sales tax), from stores or mail-order discount dealers or clubs; that price must include all the important features that make the program popular, so that you don't have to spend money for extra-cost options; and there must be no "catch" (such as having to buy a particular brand of clone).

Of those four factors, easiness is the most important: if a word processor isn't easy enough, you'd be better off with a typewriter! Fanciness is important also: if a word processor isn't fancy enough, it's inadequate for handling advanced documents. Quickness is somewhat less important: even the slowest word processors are bearable. Cheapness is the least important, since a top-notch word-processing program is well worth paying extra money for, and since mail-order dealers sell almost all those word processing programs for under $250.

In that chart, the "weighted average" comes from adding 8 times the Easy score, 6 times the Fancy score, 2 times the Quick score, and 1 times the Cheap score, then dividing the sum by 17.

As you can see from the weighted average, no program's been invented yet that deserves a straight A or A+. I hope somebody invents one!

The best so far is Word Perfect 4.1, which gets an A-. Then comes Writing Assistant, then PFS Write C, then a four-way tie for fourth place, then the rest of the gang.

In that chart, the rightmost column explains why each of those 21 programs is popular. 6 are popular because they're very fancy: they got an A or A+ for fanciness. 2 are popular because they're very easy: they got an A or A+ for easiness. 2 are popular because they're very cheap: they got an A+ for cheapness. 3 are popular because they live moderately, without "going off the deep end": they're the only programs that got at least a B in every category. The 9 remaining programs are popular because they're "historic": they appeal to old computerists who long ago learned an old-fashioned word-processing system and don't want to bother switching.

Let's examine those 21 programs in more detail, beginning with the best. . . .

Word Perfect. Word Perfect's current version (4.1) is easy to use (B). It doesn't get an A+, because the template's 40 commands are too many to make a beginner feel comfortable, the screen's right margin is occasionally incorrect (until you move the cursor), the error messages and hints on the screen are too brief to help beginners, and its whole approach to handling hyphens is annoying.

Word Perfect is amazingly fancy (A+). It even includes a thesaurus. It handles footnotes, the table of contents, the index, and its built-in thesaurus in ways that are top-notch, although its search-and-replace routine doesn't handle capitalization or punctuation well enough.

It's quick (A) but not A+: when you tell it to move the cursor down it reacts slowly, since it's trying to also adjust the line's right margin.

It's somewhat expensive (C). It lists for $495; discount dealers sell it for $197.

Writing Assistant. Of all the popular word-processing programs, Writing Assistant is the easiest (A+). It's also reasonably quick (B) and cheap (B): it lists for just $149. Since discount dealers rarely carry it, you'll probably have to buy it at your local retail store and pay the full $149. (IBM used to give it away for free, but that promotion has ended.)

Its only flaw is that it's not fancy enough. It can't really right-justify; it won't let you create a narrow paragraph (whose margins are indented from the other paragraphs'); it doesn't include any commands to help you print subscripts or superscripts or footnotes or a table of contents or an index. Although its spelling checker is nice and its search-and-replace command is quite flexible and fancy, Writing Assistant's overall fanciness score is a D, which is unacceptable.

The current version (1.01) messes up the right margin when you delete a block, although the previous version (1.0) didn't have that bug.

PFS Write. PFS Write is almost the same as Writing Assistant.

Version A of PFS Write was invented by Software Publishing Company (SPC). Then IBM hired SPC to develop an improvement, called Writing Assistant. Then SPC developed a further variation, called PFS Write Version B, which is fancier than Writing Assistant: PFS Write Version B can right-justify and also handle narrow paragraphs. Then SPC developed an improvement, called PFS Write Version C, which can handle italics, subscripts, and superscripts.

Unfortunately, the template used by PFS Write Versions B and C mimic the old template of PFS Write Version A. That template isn't as easy for a beginner as the template that IBM copyrighted for Writing Assistant. IBM's template provides an easier way to move and delete blocks.

PFS Write Version C lists for $140, but discount dealers sell it for $75. Like IBM Writing Assistant and Word Perfect, PFS Write Version C includes a spelling-checker.

PFS Write Version C works well with a data-management system called *PFS File*. Get PFS Write Version C if you want to write personalized form letters to people whose names were stored using PFS File.

Samna. Samna Word 3 (revision 3.0) is very similar to Word Perfect.

In some ways, Samna Word 3 is easier than Word Perfect: Samna's template is less intimidating, Samna displays more messages explaining what to do next, and Samna handles hyphens and the right margin more naturally, with fewer hassles.

On the other hand, Samna's method of inserting is more tedious. Before you insert some text, you must get into "insert mode", by hitting the Ins key; when you've finished inserting, you must get out of insert mode, by hitting the Ins key again. (By contrast, Word Perfect lets you insert without changing modes or hitting keys.) Moreover, while you're using Samna's "insert mode", the up-arrow and down-arrow keys don't work, and neither do most of the fancy commands, such as UNCENTER. Also, after you insert text, the page numbers on the screen are wrong, until you give a REPAGE command.

Another nuisance is that Samna forces you to invent a name for each document that's over a page long or that you wish to edit heavily, even if you don't want to save the document or use it again.

So in some ways, Samna is easier than Word Perfect, and in other ways, Samna is harder. It winds up with the same score for easiness: B.

For fanciness, Samna is almost as good as Word Perfect, and so it gets an A (though not an A+). Though Samna handles search-and-replace better than Word Perfect, Samna's method of indexing is inferior, and Samna lacks a thesaurus.

After Samna Corporation receives your warranty card, the Corporation will answer all your technical questions, on its toll-free WATS line, for a month. After the month ends, Samna Corp. charges you about $12 per extra phone call. (Word Perfect's publisher, by contrast, gives you free help forever.)

Leading Edge. The Leading Edge Word Processor is sold by Leading Edge Products Inc., which also sells the Leading Edge Computer. If you buy the Leading Edge Computer (which is an IBM PC clone), you usually get the Leading Edge Word Processor free. Although it's easy to use (B), it's not quite as fancy as Word Perfect and Samna Word 3: it doesn't help you create footnotes, a table of contents, or an index. The plain version lists for $100 ($49 at discount); the fancy version, which includes a spelling checker, lists for $250 ($99 at discount). If you buy a Leading Edge computer (Model D or Model M), you get the plain version free.

Microsoft Word. Microsoft Word 2 is the only word processor that can take full advantage of fancy hardware.

For example, if you buy Hewlett-Packard's fancy Laserjet Printer for your IBM PC, Microsoft Word will make the Laserjet print very beautifully, using many typestyles. If you buy a Hercules card (which lets your IBM monochrome monitor handle graphics), Microsoft Word will let your monitor display italics and super-small characters—so small that you can see 90 of them per line, and 43 lines per screen. If you buy Microsoft's mouse (which you roll across your desk to move the cursor), Microsoft Word will let you use the mouse instead of the arrow keys.

But though the mouse is fun and cute, moving your hands from the keyboard to the mouse slows you down. On the other hand, if you *don't* use a mouse, Microsoft Word requires too many keystrokes.

Also, Microsoft Word runs too slowly: you must often wait for the computer to catch up with you.

Though Microsoft Word includes a disk that contains a cute tutorial, that disk and the accompanying manual teach you too slowly. When you try to look up a topic in the manual, you'll find the manual resembles a politician: it rambles on at great length, but ducks answering your question directly. It keeps telling you to try the tutorial disk instead.

Microsoft Word is also available for the Macintosh. It fits the Macintosh better than it fits the IBM PC, because the typical Macintosh keyboard lacks arrow keys and so *needs* a mouse-oriented word processor such as Microsoft Word.

Xywrite. Of all the word processors that are fancy, the only one that's amazingly quick is Xywrite (pronounced "zai write"). Unfortunately, it doesn't offer an easy menu or template: you must memorize its commands, which are hard to learn.

Xywrite was invented by former employees of Atex, a typesetting company. Xywrite is particularly popular among authors who want to send books to typesetters, because Xywrite lets you easily insert typesetter codes in the middle of your document and make those codes invisible, so that your screen remains uncluttered.

Wordstar 2000. Wordstar 2000 Plus is fairly easy (B). It's also quite fancy (A), though it can't handle footnotes. Unfortunately, it's so slow that you'll need to buy a hard disk, which works faster than a floppy disk.

Wordstar 2000 Plus lists for $595 ($278 at discount). A stripped down version, called just Wordstar 2000, lists for $495 ($233 at discount). The Plus version has three advantages over the plain version: the Plus version can generate an index, handle mailing lists, and help transmit documents over the phone.

Volkswriter. Volkswriter was the first practical, easy-to-use word processor invented especially for the IBM PC. It was the first word processor to make good use of the ten function keys (F1 through F10). It was invented by a group of California hippies who called themselves Lifetree Software and loved Volkswagens. The latest version—Volkswriter 3— resembles the original but is fancier.

After Volkswriter was invented, competitors arose that were even easier (such as Writing Assistant) and that made even fancier use of the function keys (such as Word Perfect). But Volkswriter 3 is a pleasant compromise between the simplicity of Writing Assistant and the fanciness of Word Perfect. Its price is intermediate also: it lists for $295; discount dealers charge $142. Moreover, it's particularly good at handling European accents, and you can get a special version (called Volkswriter Scientific) that handles mathematical symbols and formulas.

The previous version of Volkswriter couldn't handle insertions well: after inserting a phrase, you had to give a REFORM command to adjust the paragraph's right margin. The REFORM command adjusted just one paragraph at a time: if you inserted three paragraphs, you had to move the cursor back to each paragraph and give the REFORM command for each paragraph, three times altogether.

Volkswriter 3 solves that problem—partly. It automatically REFORMs each paragraph—but it sometimes accidentally puts extra spaces near periods.

Einstein Writer. Einstein Writer's a fine word processor that lists for just $170. But since it doesn't do anything outstanding or unusual, everybody ignores it. Discount dealers have stopped bothering to advertise it, and it will probably fade away.

Textra. Textra 3.1 lists for just $9.95. Yes, that low price gives you the disk and the instruction manual. Get Textra 3.1 directly from the publisher, Ann Arbor Software (407 North Main St., Ann Arbor, MI 48104, phone 313-769-9088). After buying it, you'll also want to buy a supplementary disk, called Textra 3.1A, which adds 50 more features and costs $14.95. So altogether, for a total of $24.90, you get the two disks and their two manuals, and you have altogether a nice word processing system. It also runs amazingly fast (A+).

PC-Write. Like Textra, PC-Write is amazingly fast and amazingly cheap. You get PC-Write for just $10 from the publisher, Quicksoft (219 First N. #224, Seattle, WA 98109, phone 206-282-0452). That low price doesn't include a printed manual, but the disk includes software that makes your own printer print the manual for you!

The publisher lets you copy PC-Write for free and give it to all your friends. So before buying PC-Write, ask your friends and local computer club whether they've bought the program already, so that you can copy *their* disk. Since PC-Write is software that you're allowed to share with your friends, it's called *shareware*.

After you've tried the program and enjoyed it, the publisher suggests that you support for the shareware concept by sending the publisher a $75 donation. In return, the publisher will send you an instruction manual that's beautifully bound, two disks full of extra goodies, and several issues of the publisher's newsletter.

If you convince your friends to donate also, the publisher will give you a $25 commission for each friend who donates $75.

Wordstar. Wordstar was the first fancy word-processing program for microcomputers. At first, it ran just on ancient computers that used the CP/M operating system. Later, Wordstar's publisher (Micropro) rewrote it to run on the IBM PC.

Unfortunately, Wordstar's commands are hard to remember, though Wordstar tries to help you by continually printing menus of them on your screen. Another disadvantage of Wordstar is that insertions wreck the right margin, until you give a REFORMAT command. Though many folks still use Wordstar, and though it's been gradually improved over the years, it's not yet as pleasant as its competitors.

The current version, Wordstar 3.3, costs $350 list, $162 at discount. A slightly souped-up version, called Wordstar Professional, adds spelling correction, index generation, and hookups to data-management systems; it costs $495 list, $233 at discount.

To rebut criticism of Wordstar 3.3 and Professional, Micropro has developed Wordstar 2000, which is quite different and superior. I gave it a separate line in the chart.

Others. *Easy* is a stripped-down version of Wordstar. It's easier to learn and costs less.

Office Writer, *Multimate Professional*, *Multimate Advantage*, and *Just Write* all make the IBM PC imitate a Wang Word Processor. Office Writer is the easiest and quickest; Just Write is the cheapest; Multimate Professional is the oldest and most famous; and Multimate Advantage is the "improved" version of Multimate (and therefore more expensive and less of a bargain).

Unfortunately, the Wang Word Processor that they imitate is so outmoded that it isn't worth imitating. Its screen can't simultaneously show the bottom of page 1 and the beginning of page 2, and it hassles you when you try to move information from one page to another.

Wang imitations appeal only to people who already learned how to use a Wang Word Processor and want the IBM PC to imitate the nightmare they've already learned.

To compete against the Wang Word Processor, IBM invented a machine called the *Displaywriter*, which wasn't much better, and which IBM has stopped selling. Now IBM tells customers to buy an IBM PC plus a program called *Displaywrite 3*, which makes the PC imitate the Displaywriter. IBM must be a sadist! Moreover, IBM charges $450 for Displaywriter 3, and you can't get discounts, except when IBM gives it away "free" to fools who pay list price for IBM's computers.

Bank Street Writer was designed for kids using an Apple 2+. Since that program was so easy, adults began using it too. Then its publisher, Broderbund, developed versions of it for the Apple 2e & 2c, the Commodore 64, and the IBM PC. Though easy to use, its popularity is fading, because newer programs (such as Writing Assistant) are even easier.

Homeword was invented because the president of Sierra On-Line publishing company, Ken Williams, bought a Porsche. He noticed that the Porsche's fancy high-tech dashboard used easy-to-understand pictures (called *icons*) instead of words. Inspired by the Porsche, his company invented Homeword—the first IBM PC word processor that used icons instead of commands. It made the IBM resemble a Macintosh. IBM marketed Homeword as being the ideal word processor for kids using the PC Junior. But since the PC Junior didn't become popular, neither did Homeword. Later, competition from Writing Assistant killed Homeword: discount dealers have stopped trying to sell the IBM version of Homeword, although the Apple, Commodore, and Atari versions are still popular.

WORD PROCESSORS FOR OTHER COMPUTERS

The best word-processing programs require an IBM PC or clone. If you don't have an IBM PC, here's what to get instead. . . .

For the Commodore 64, get *Omniwriter*. For most Atari computers, get *Atariwriter*. For the Apple 2e & 2c, get *Word Perfect* or *PFS Write*, which have been translated from the IBM PC. (The Apple versions cost less than the IBM PC versions and aren't as fancy.) For the Macintosh, get *Macwrite* (which is easy) or *Microsoft Word* (which is fancier than Macwrite but harder).

For Radio Shack models 3, 4, 4P, and 4D, get *Super Scripsit*, which is the word-processing program I used for writing this book; an alternative is plain *Scripsit*, which is easier but not as fancy. For the Radio Shack Color Computer 1 and 2, get *Telewriter 64*, which is sold by independent dealers and better than the "Color Scripsit" that your local Radio Shack store sells.

For CP/M computers, the easiest word-processing program is *Select*. *Wordstar* is much harder, but popular because it often comes free with the computer and performs fancy tricks. *Newword* is a slight improvement of Wordstar.

WOW! YOU FINISHED CHAPTER 7: WORD PROCESSING! TURN THE PAGE FOR CHAPTER 8: SPREADSHEETS.

BOP AROUND

KINDS OF SPREADSHEETS

A *spreadsheet* is any table of numbers. The numbers in the table can be how many dollars you earned or spent or plan to spend, or how much inventory you have in stock, or what your students scored on tests, or how the public responded to a survey, or measurements from a scientific experiment, or whatever other numbers interest you!

For example, this spreadsheet deals with money:

	January	February
Income	$9,030.95	$12,486.99
Expenses	$7,000.55	$9,210.75
Profit	$2,030.40	$3,276.24

A *spreadsheet program* helps you create a spreadsheet, edit the spreadsheet, and analyze it.

The first spreadsheet program was invented by Dan Bricklin and Bob Frankston in 1979. They called it *Visicalc*, because it was a "visible calculator".

Later, other programmers invented an improved version, called *Supercalc*.

Lotus 1-2-3. Today, the most popular spreadsheet program is the one invented by Mitch Kapor and Jonathan Sachs. They call it *1-2-3*, because it runs quickly and can handle three topics: spreadsheets, graphics, and data-management. It's published by Mitch's company, which he calls *Lotus Development Corporation*, because he also teaches transcendental meditation.

Lotus shipped the first copies of 1-2-3 to dealers at the beginning of 1983. Lotus shipped about a half million copies of it in 1983, about a half million more copies of it in 1984, and even more copies since then.

The original version of 1-2-3 was called *Release 1*. Then came an improvement, called *Release 1A*. But Release 1 and Release 1A are no longer sold, because they've been replaced by a further improvement, called *Release 2*, which lists for $495. Many discount dealers sell it for just $299, in spite of Lotus's attempts to punish them.

Clones of 1-2-3. If you can't afford even $299, cheer up: you can buy an imitation of 1-2-3 for much less!

The most popular imitations are *Farsight* ($99), *Twin* ($99 list, $72 at discount), and *VP Planner* ($99 list, $49 at discount). They imitate Release 1A, though not Release 2. They run slower than 1-2-3, come with worse manuals, can't handle as great a variety of printers, and aren't quite perfected yet; but at those low prices, they're bargains! Moreover, they include some extra commands that 1-2-3 lacks: Farsight includes extra commands for word processing, Twin includes extra commands for graphics, and VP Planner includes extra commands for data management.

What's best? The best spreadsheet programs are 1-2-3 and *Excel*. 1-2-3 runs only on the IBM PC (and clones); Excel runs only on the Macintosh. Excel is published by Microsoft and sells for $395 list, $221 at discount.

Lotus, which publishes 1-2-3, also publishes *Symphony* and *Jazz*. They're more eye-catching than 1-2-3, but Symphony is too complicated and Jazz is too superficial. Symphony runs on the IBM PC; Jazz runs on the Macintosh.

If you don't have an IBM PC or clone or Macintosh, you must settle for a spreadsheet program that's "second-best". If you have an Apple 2e or 2c, get *Supercalc 3A* ($101 at discount). *Multiplan* is popular because it runs on the greatest variety of computers: it runs on the IBM PC, Apple 2e & 2c & Macintosh, Commodore 64, and Radio Shack Model 4. *Visicalc* is still available for IBM and Radio Shack computers, though the Apple version has been discontinued.

PFS Plan, *Planning Assistant*, and *Javelin* are unusual spreadsheet programs that let you more easily manipulate whole rows and columns, but on the other hand prevent you from easily manipulating individual entries.

Emphasis. Since 1-2-3 is the most popular spreadsheet program, this chapter explains how to use the current version of 1-2-3: Release 2. It also makes comments about the preceding version (Release 1A).

HOW TO START USING 1-2-3

When you buy 1-2-3 Release 2, you get six disks. The 1-2-3 System Disk is the main disk: it contains 1-2-3 itself. The Backup System Disk is a copy of the 1-2-3 System Disk, in case the System Disk gets damaged. The Print Graph Disk helps you print graphs onto paper. The View Disk gives beginners a preview of what 1-2-3 is all about. The Utility Disk and the Install Library Disk help customize 1-2-3 for your own printer and monitor. (Release 1A lacks the Install Library Disk.)

Put the PC-DOS System Master Disk into drive A, turn on the computer, and type the date and time. Put the 1-2-3 System Disk in drive A; and after the "A>", type the word "lotus". (Exception: if your 1-2-3 System Disk is new and nobody's used it yet, you must *install* it before using it. To find out how to install it, read Lotus's manual.)

The computer will put this menu on the screen:

1-2-3 Print-graph Translate Install View Exit

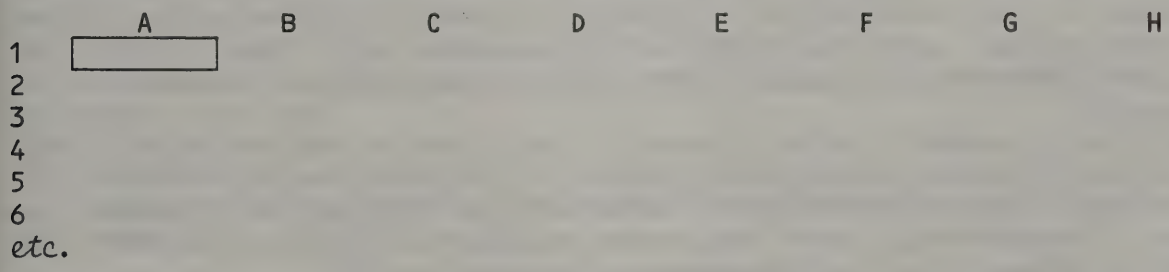
(Release 1A uses slightly different words.)

To choose an item from that menu, type the first character of the item's name. For example, to choose 1-2-3, tap the 1 key; to choose Print-graph, tap the P key; to choose Translate, tap the T key.

If you're not sure which item to choose, the computer will help you decide. Notice the box around 1-2-3. While the box surrounds 1-2-3, the computer's screen shows a brief explanation of what 1-2-3 means. If you tap the right-arrow key, the box moves to the right, so that it surrounds Print-graph, and computer explains what Print-graph means. If you tap the right-arrow key again, the box moves farther to the right, so that it surrounds Translate, and the computer explains what Translate means.

To begin wisely, choose 1-2-3 from the menu. To do that, tap the 1 key.

The computer will print a message, reminding you that 1-2-3 is copyright by Lotus Development Corporation. (If you're using Release 1A, press the ENTER key afterwards.) Then your screen will look like this:



MOVING THE BOX

Notice that the box is in column A, row 1. If you tap the right-arrow key, the box will move to the right, so that it's in column B. If you tap the down-arrow key, the box will move down, to row 2. By tapping the four arrow keys, you can move the box in all four directions, so that you can move the box practically anywhere on the screen. If you press the HOME key, the box will hop back to where it began: column A, row 1.

Each possible position of the box is called a *cell*. When you begin using 1-2-3, the box is in column A, row 1; that position is called *cell A1*. If you move the box from there to the right edge of the screen, the box will be in column H, row 1; that position is called *cell H1*.

To create a spreadsheet, move the box from cell to cell, and put into each cell whatever words or numbers you wish.

YOUR FIRST SPREADSHEET

Let's create this spreadsheet:

	A	B	C	D	E	F	G	H
1		January						
2	Income	7000						
3	Expenses	5000						
4	Profit	2000						
5								
6								

etc.

To begin, move the box to cell A2, then type the word Income. (While you're typing the word Income, it will temporarily appear at another place, called the *edit line*, but it will go into cell A2 as soon as you press the ENTER key or an arrow key.)

Then move the box to cell A3; type the word Expenses. Move the box to cell A4; type the word Profit. Move the box to cell B1; type the word January. Move the box to cell B2; type the number 7000. Move the box to cell B3; type the number 5000.

While you're typing those words and numbers, you can use the BACKSPACE key to erase the last character you typed. The left-arrow key will *not* erase the last character you typed; instead, the left-arrow key moves the box to a different cell.

Although the computer's screen shows the words you typed (Income, Expenses, and Profit), the computer doesn't understand what those words mean. It doesn't know that "Profit" means "Income minus Expenses". The computer doesn't know that the number in cell B4 (which represents the profit) ought to be the number in cell B2 (the amount of income) minus the number in cell B3 (the dollars spent).

You must *teach* the computer the meaning of Profit, by teaching it that the number in cell B4 ought to be the number in cell B2 minus the number in cell B3.

To do that, move the box to cell B4, then type:

```
+B2-B3
```

That formula begins with a plus sign, because *every* formula should begin with a plus sign. The rest of the formula, B2-B3, tells the computer to subtract the number in cell B3 from the number in cell B2, and put the answer into the box's cell (which is cell B4). You don't have to bother capitalizing the B: capitalization is optional.

While you're typing that formula, it appears in the edit line. When you've finished typing that formula, press the ENTER key (or an arrow key). Then the computer automatically computes the formula's answer (2000) and puts that number into the box's cell (B4).

That formula remains in effect forever. It says that the number in cell B4 will always be the B2 number minus the B3 number. If you ever change the numbers in cells B2 and B3 (by moving the box to those cells and retyping the numbers), the computer will automatically adjust the number in cell B4, so that the number in cell B4 is still B2 minus B3 and still represents the correct profit.

For example, suppose you move the box to cell B2, then type 8000 (to change the January income to $8000), and then press the ENTER key. As soon as you press that ENTER key, not only does the number 8000 appear in cell B2 but also the profit in cell B4 immediately changes to 3000, right in front of your eyes!

A typical spreadsheet contains *dozens* of numbers, totals, subtotals, averages, and percentages. Each cell that contains a total, subtotal, average, or percentage is defined by a formula. Whenever you retype one of the numbers in the spreadsheet, the computer automatically readjusts all the totals, subtotals, averages, and percentages, right in front of your eyes.

Remember to begin each formula with a plus sign. Inside the formula, you can use the same math symbols that BASIC uses:

```
+   -   *   /   (   )
```

MISTAKES IN FORMULAS

If you make a mistake in a formula (such as typing an X instead of *), when you press the ENTER key the computer will gripe, by beeping at you.

After the computer beeps, you can correct the error in two ways.

The simplest is to hold down the CONTROL and BREAK keys simultaneously; that makes the computer erase the entire formula and let you start over. (The CONTROL key says "Ctrl" on it; the BREAK key is in the upper-right corner and says "Scroll Lock Break".)

A fancier way to correct the error is to move the cursor to the bad character (the X), by pressing the left-arrow key. Then delete that character, by tapping the DELETE key, which says "Del" on it. Then type the character you want to insert (the *). Then press the ENTER key.

FUNCTIONS

The computer understands the word *sum*. For the sum of cells B2 through B9, instead of typing +B2+B3+B4+B5+B6+B7+B8+B9, you can type:

```
+@sum(B2...B9)
```

Notice that the word sum must be preceded by the symbol @ and followed by parentheses.

When typing that formula, you can abbreviate. Instead of typing three periods, you can type just one, like this:

```
+@sum(B2.B9)
```

If a formula begins with the symbols +@, you can omit the plus sign. So you can type just:

```
@sum(B2.B9)
```

If you type @sum(B2.H2), the computer will find the sum of cells B2 through H2, which is +B2+C2+D2+E2+F2+G2+H2. If you type @sum(B2.C4), the computer will find the sum of all the cells in the rectangular area that stretches from B2 to C4; the computer will do +B2+B3+B4+C2+C3+C4.

To find the average of cells B6 through B9, you can type +(B6+B7+B8+B9)/4, but a shorter way is to type:

```
@avg(B6.B9)
```

To find the average of cells C7, B5, and F2, you can type +(C7+B5+F2)/3, but a nicer way is to type:

```
@avg(C7,B5,F2)
```

The symbols @sum and @avg are both called *functions*. Each function begins with the symbol @.

POINTING

While you're typing a formula, you can point to cells instead of typing their names. For example, in the middle of a formula, instead of typing B2, you can just point to cell B2, by moving the box to cell B2. When you move the box to cell B2, the computer automatically types B2 for you.

So to type the formula +B2-B3, just type the plus sign, then move the box to cell B2, then type the minus sign, then move the box to cell B3.

So to find the sum of all the cells from B2 to B9, for which the formula is @sum(B2.B9), type the @sum(, then move the box to cell B2, then type a period, then move the box to cell B9, then type the final parenthesis.

In a large spreadsheet, if you want the number in one cell to be the sum of numbers in two other cells, it's faster to point to the two other cells (by moving the box to them) than to type those cells' names, because to type them you'd have to run your eye across the whole spreadsheet to find the name of each cell's column and the number of each cell's row. Conclusion: to save time, always point to cells instead of typing their names.

CENTERING

When you type a word (such as January) and then press the ENTER key, the computer puts the word into a cell. Each cell is 9 characters wide. If the word is shorter than 9 characters, the computer puts the word near the cell's left edge, and enough blank spaces after the word to fill up the cell. For example, if you type the word January, the cell will contain the word January followed by 2 blank spaces.

If you want a word to be *centered* in the cell, type the symbol ^ before typing the word. For example, if you want the word January to be centered in the cell, type ^January and then press the ENTER key. When you press the ENTER key, the computer will put January into the cell, and will put a space before and after January, so that January is centered.

For another example, if you type ^May and then press the ENTER key, the computer will put May into the cell, and will put 3 blank spaces before and after May, so that May is centered.

If you type a quotation mark instead of a caret, the computer will put May into the cell and near the cell's right edge (instead of centered), so that May has just one space after it (and 5 spaces before it).

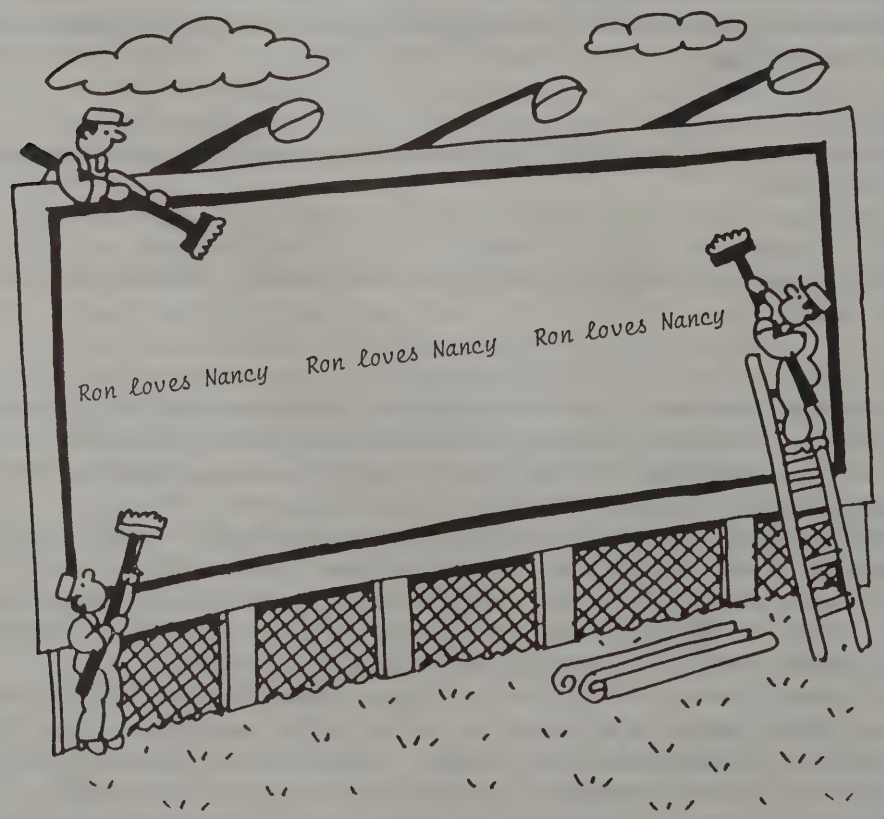
FILLING

Suppose you want to fill an entire cell with dashes, so that the cell looks like this:

To do that quickly, use this short-cut: type a *backslash* (which is the symbol "\"), then type the dash, then press the ENTER key. The backslash means: fill. So the computer will fill the entire cell with dashes.

To fill a cell with JOY, type a backslash, then type JOY, then press the ENTER key. The computer will make the cell look like this:

JOYJOYJOY



SLASH YOUR PROBLEMS

WORKSHEET ERASE YES

To erase the entire worksheet, so that all the cells are blank and you can start over again, give the Worksheet Erase Yes command. Here's how. . . .

To give a command, type a slash (which is the symbol "/"), then type the first letter of each word in the command. So to type the Worksheet Erase Yes command, type a slash, then a W, then an E, then a Y.

TRICKS WHEN GIVING COMMANDS

When typing a command, don't bother capitalizing the letters; don't bother pressing the SHIFT key. So to type the Worksheet Erase Yes command, just tap the slash key, then the W key, then the E key, then the Y key.

As you tap each of those keys, the screen displays a message reminding you what to type next. For example, after you tap the slash key, the screen shows a message reminding you that W (for Worksheet) can be typed next.

Instead of Worksheet, you can give one of these commands instead: Range, Copy, Move, File, Print, Graph, Data, System, or Quit. I'll explain those alternatives later. The message on the screen reminds you of those alternatives also. So after you tap the slash key, the screen shows this menu:

```
Worksheet  Range  Copy  Move  File  Print  Graph  Data  System  Quit
```

(Exception: Release 1A doesn't include System.)

After you choose Worksheet from that menu (by tapping the W key), you see another menu, which includes Erase. After you say Erase (by tapping the E key), you see another menu, which includes Yes. After you say Yes (by tapping the Y key), the computer obeys your command and erases the entire worksheet.

While the computer displays a menu, it also tutors you, by displaying an explanation of one of the items in the menu. To see an explanation of a different item in the menu, tap the right-arrow key. By tapping the right-arrow key several times, you can explore the meaning of each item in the menu. When you finally decide which item in the menu interests you, type the first letter of that item's name.

How to cancel. Suppose you start typing a command (and see a menu on the screen) but then change your mind. To cancel your last keystroke, tap the ESCAPE key (which says "Esc" on it).

To cancel the entire command (all your keystrokes, and even the slash), you can tap the ESCAPE key several times. A shorter way to cancel the entire command is to tap the BREAK key while holding down the CONTROL key.

RANGE ERASE

Here's how to erase just a few of the cells.

Move the box to the first cell you want to erase, then say Range Erase (by tapping the slash key, then the R key, then the E key). Move the box to the last cell you want to erase, then press the ENTER key.

The computer will erase that first cell and last cell and the cells between them.

For example, if the first cell is B2, and the last cell is B7, the computer will erase B2, B3, B4, B5, B6, and B7. If the first cell is B2, and the last cell is E2, the computer will erase B2, C2, D2, and E2. If the first cell is B2, and the last cell is C4, the computer will erase B2, B3, B4, C2, C3, and C4. If the first cell is B2, and the last cell is also B2, the computer will erase just B2; that's how you make the computer erase one cell.

The first cell, the last cell, and any cells between them form the *range*. So to erase a range, point to the first cell in the range (by moving the box), then say Range Erase; point to the last cell in the range, then press the ENTER key.

WORKSHEET DELETE COLUMN/ROW

To delete several columns, point to the first column you want to delete (by moving the box into that column), then say Worksheet Delete Column. Point to the last column you want to delete, then press the ENTER key.

The computer will delete those columns and all the columns between them. The computer will also rearrange the spreadsheet, to close the gap left by the deleted columns.

For example, suppose you say to delete columns C, D, and E. The computer will erase all the data from columns C, D, and E. Into the space left by the missing data, the computer will move the data from columns F, G, and H. At the top of the screen, you'll still see all the letters (A, B, C, D, E, F, G, and H); but now columns C, D, and E contain the data that used to be in columns F, G, and H.

When the computer rearranges the spreadsheet, it also adjusts each formula. For example, since the data from column F has moved to column C, the computer hunts through all formulas in the spreadsheet, and adjusts them by changing each "F" to "C" (and each "G" to "D" and each "H" to "E").

To delete rows instead of columns, point to the first row you want to delete, then say Worksheet Delete Row. Point to the last row you want to delete, then press the ENTER key.

WORKSHEET INSERT COLUMN/ROW

To insert extra columns in the middle of your spreadsheet, point to where you want the first inserted column to be, then say Worksheet Insert Column. Point to where you want the last inserted column to be, then press the ENTER key.

The computer will move other columns out of the way, to make room for the new columns. The computer will also adjust each formula.

To insert extra rows in the middle of your spreadsheet, point to where you want the first inserted row to be, then say Worksheet Insert Row. Point to where you want the last inserted row to be, then press the ENTER key.

MOVE

Here's how to move a range of cells to a different part of the spreadsheet.

Point to the beginning of the range, then say Move (by tapping the slash key then the M key). Point to the end of the range, then press the ENTER key. Where do you want to move the beginning of the range? Point there, then press the ENTER key again.

The computer will move the beginning of the range to where you pointed. It will move the rest of the range also, so that the range remains together. It will also adjust all formulas that refer to the moved range.

Unfortunately, the computer will *not* move any cells out of the way, to make space for the range's new location. So before saying Move, make sure the place you're moving to is blank space. (If it's *not* blank, create new blank columns or rows, by saying Worksheet Insert Column or Worksheet Insert Row.)

When the computer finishes the move, it does *not* close the gap where the range used to be. If you want to close that gap, point there and say Worksheet Delete Column or Worksheet Delete Row.

COPY

Here's how to copy a range of cells to a different place, so that the range will appear in *both* places.

Point to the beginning of the range, then say Copy (by tapping the slash key then the C key). Point to the end of the range, then press the ENTER key. Where do you want to copy the beginning of the range? Point there, then press the ENTER key again.

The computer will copy the beginning of the range to where you pointed. It will copy the rest of the range also, so that the range remains together.

The computer will *not* move any other cells out of the way, to make room for the new copy. So before saying Copy, make sure the place you're copying to is blank space. (If it's *not* blank, create new blank columns or rows, by saying Worksheet Insert Column or Worksheet Insert Row.)

Copying formulas. If one of the cells you're copying contains a formula, the computer does *not* copy the formula exactly; instead, it copies the relationship that the formula represents.

For example, suppose cell C4 contains the formula +C2+C3, so that C4 is the sum of the two cells above it. If you tell the computer to copy cell C4 to F9, the computer will make F9's formula be "the sum of the two cells above it"; the computer will make F9's formula be +F7+F8.

For another example, suppose cell C4 contains the formula +2*C3, so that C4 is twice the cell above it. If you tell the computer to copy cell C4 to F9, the computer will make F9's formula be "twice the cell above it"; the computer will make F9's formula be +2*F8.

For another example, suppose cell C4 contains the formula +2*B4, so that C4 is twice the cell to the left of it. If you tell the computer to copy cell C4 to F9, the computer will make F9's formula be "twice the cell to the left of it"; the computer will make F9's formula be +2*E9.

Dollar signs. Notice again how copying from C4 to F9 turns the formula +C2+C3 into +F7+F8: it turns each C into an F, the 2 into a 7, and the 3 into an 8. If you want to prevent those changes, put dollar signs in the original formula. For example, to prevent the 3 from turning into an 8, put a dollar sign before the 3, so that cell C4 contains this:

```
+C2+C$3
```

When you copy that cell to F9, the dollar sign prevents the computer from turning the 3 into an 8; F9's formula will become +F7+F$3 (instead of +F7+F8).

To prevent the 2 from turning into a 7, put a dollar sign before the 2, like this:

```
+C$2+C3
```

To prevent each C from turning into an F, put a dollar sign before each C, like this:

```
+$C2+$C3
```

To prevent any changes from occurring at all, put a dollar sign before each column letter and each row number, like this:

```
+$C$2+$C$3
```

When you copy that formula from cell C4 to F9, the computer will put that same formula into cell F9, without making any changes. Cell F9 will contain +$C$2+$C$3.

Instead of typing $C$2, you can point to cell C2 and then press the F4 key. Pressing the F4 key makes the computer insert the dollar signs. So to type +$C$2+$C$3 quickly, just tap the + key, then point to cell C2, then tap the F4 key, then tap the + key again, the point to cell C3, then tap the F4 key again.

Kinds of addresses. A cell's name (such as C3) is called the cell's *address*, because the cell's name tells you where to find the cell.

An address that contains dollar signs (such as $C$3) is called an *absolute address*, because the address is absolutely fixed and will never change, not even when you copy the formula.

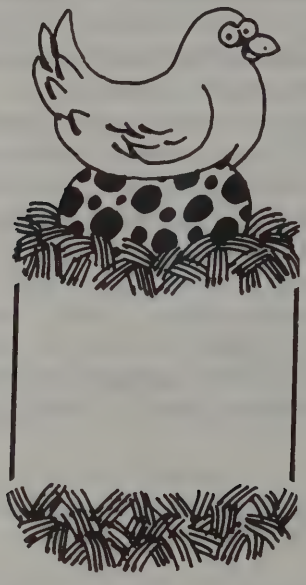
An address that lacks dollar signs is called a *relative address*, because when you copy that address you'll be copying the cell's relationship to the other cells.

An address containing just one dollar sign (such as C$3) is called a *mixed address*, because it's partly relative and partly absolute.

Multiple copies. Suppose C3, C4, and C5 contain formulas. Let's copy those formulas (or the relationships they represent) to column D, and also copy them to column E, and also to column F, and also to column G.

Instead of giving the Copy command 4 times, you can perform one "super" Copy that does it all! Here's how. . . .

Point to C3 (which is the beginning of the range), then say Copy. Point to C5 (which is the end of the range), then press the ENTER key. Point to D3 (which is where you want the beginning of the first copy to appear), then press the period key. Point to G3 (which is where you want the beginning of the last copy to appear), then press the ENTER key.



One Lotus Lane One Poultry Lay'n

BIG SHEETS

HOW BIG CAN YOUR SPREADSHEET GET?

When you're using 1-2-3, your screen normally shows 20 rows (numbered from 1 to 20). It normally shows 8 columns (lettered from A to H), and each column appears 9 characters wide.

But 1-2-3 can handle spreadsheets much bigger than that. Release 1A can handle 2048 rows, and Release 2 can handle 8192! Either release can handle handle 256 columns, lettered from A to Z, then from AA to AZ, then from BA to BZ, and so on, up to IV. (That's why people who use 1-2-3 are called "IV league accountants".)

Moreover, 1-2-3 can handle spreadsheets that have wide columns: up to 240 characters per column!

Unfortunately, a spreadsheet that contains 8192 rows with 256 columns per row and 240 characters per column will require 480 *mega*bytes of RAM; but the IBM PC normally comes with just 256 *kilo*bytes. Although you can easily expand the IBM PC's RAM to 640 kilobytes, and you can expand it to 4¼ megabytes by getting a high-priced memory board, you're still nowhere near the 480 megabytes that the maximum spreadsheet requires. Cynics say, "Before you run out of spreadsheet, you'll run out of RAM."

EXTRA ROWS

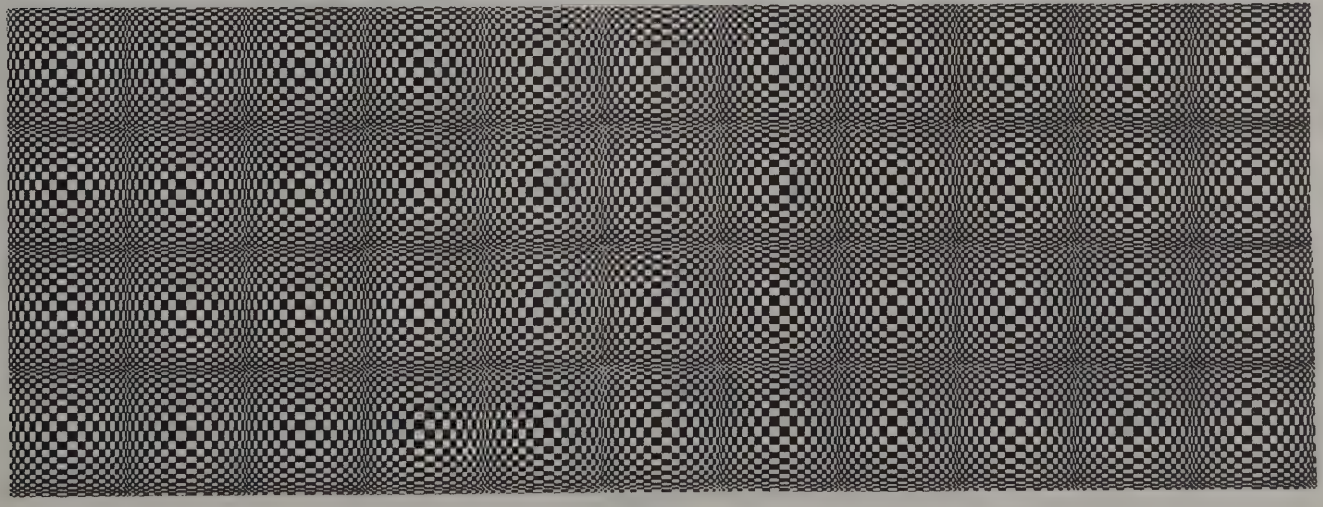
When you start using 1-2-3, your screen shows 20 rows, numbered from 1 to 20. If you press the PAGE DOWN key (which says "PgDn" on it), the screen will show rows 21 through 40. If you press the PAGE DOWN key again, the screen will show rows 41 through 60. To return to rows 1 through 20, press the PAGE UP key several times, or press the HOME key (which returns the box to cell A1).

Another way to reach row 21 is to move the box down to row 20, and then press the down-arrow key once more.

EXTRA COLUMNS

When you start using 1-2-3, your screen shows 8 columns, lettered from A to H. If you press the TAB key (which is next to the Q key), the screen will show columns I to P. If you press the TAB key again, the screen will show columns Q to X. To return to columns A to H, tap the TAB key several times *while holding down the SHIFT key*, or press the HOME key.

Another way to reach column I is to move the box across to column H, and then press the right-arrow key once more.



CHANGING THE COLUMN WIDTH

Although each cell can hold 240 characters, 1-2-3 shows you just part of each cell. When you start using 1-2-3, you see just the 9 most important characters from each cell.

Here's how to see 15 characters (instead of 9) from each cell in column E, so that column E looks 15 characters wide on your screen.

Point to column E, then say Worksheet Column Set-width (by tapping the slash then W then C then S). Type 15, then press the ENTER key.

Column E, which was 9 characters wide, suddenly expands to be 15 characters wide. Columns F and G move out of the way, to make room for it.

Column H tries to move out of the way too; but since it was already at the right edge of the screen, it can't find a place to move to, so it disappears from the screen. (If you ever want to see it again, you can reach it by pressing the right-arrow key several times, or by pressing the TAB key.)

You can make a column very wide or very narrow. If you try to type a long number in a column that's too narrow to display the number, the column will display asterisks instead of the number. So if you ever see lots of asterisks on your screen, it means a number is too long to fit in a column, and you should widen the column.

GLOBAL COLUMN-WIDTH

To make *all* the columns show 15 characters, say Worksheet Global Column-width, then type 15, then press the ENTER key.

The word *global* means "all", so *global column-width 15* means "all column-widths 15".

If you make column E become 20 characters wide, and then say Worksheet Global Column-width 15, most of the columns will become 15 characters wide, but column E will remain 20 characters wide, because the computer takes a global request less seriously than a request concerning a specific column.

SEEING BEYOND 72 CHARACTERS

Although you can type up to 240 characters into a cell, you can't make the cell show more than 72 of them on your screen, because your screen isn't wide enough.

But if you move the box to a cell in which you typed more than 72 characters, and then press the F2 key, the top of the screen will display some of those extra characters you typed. Move the left and right arrows to see the extra characters. When you've finished examining those extra characters, press the ENTER key.

TITLES

You should put a heading at the top of each column. For example, if column B contains financial information for January, and column C contains financial information for February, you should put the word January at the top of column B, and the word February at the top of column C. Since the words January and February are at the top of the columns, they're in row 1. They're called the *column titles*.

If row 2 analyzes Income, and row 3 analyzes Expenses, you should put the word Income at the left edge of row 2, and the word Expenses at the left edge of row 3. Since the words Income and Expenses are at the left edge of the spreadsheet, they're in column A. They're called the *row titles*.

So in a typical spreadsheet, the column titles are in row 1, and the row titles are in column A.

Unfortunately, when you move to a different part of the spreadsheet (by tapping the PAGE DOWN key or TAB key), the titles tend to disappear from the screen, and you forget the purpose of each row and column.

To force the titles to stay on the screen, move the box to cell B2, then say Worksheet Titles Both. Afterwards, even if you tap the PAGE DOWN or TAB key, the titles will stay on the screen.

After you've said Worksheet Titles Both, you can't move the box to row 1 or column A anymore. Row 1 and column A are off limits. For example, if you press the HOME key, which tries to send the box to cell A1, the box will go only as far as cell B2; it will stay in B2.

If you *want* to move the box to row 1 or column A (so that you can revise the titles), you must cancel the "Worksheet Titles Both" command. To cancel it, say Worksheet Titles Clear.

WINDOWS

Normally the screen shows columns A through H. If you press the TAB key, you'll see columns I through P instead.

Let's make the left side of the screen show columns A through D, while at the same time the right side of the screen shows columns I through L.

To do that, move the box to the middle of the screen (column E) and then say Worksheet Window Vertical. The screen splits into *two* displays. The display on the left shows columns A through D; the display on the right shows columns E through H. The displays are separated from each other by a vertical line.

Each of those displays is called a *window*, because each of those displays shows a separate view of the worksheet. The command is called Worksheet Window Vertical, because you see the worksheet through two windows separated by a vertical line.

The box begins in the left window, but you can move it to the right window by pressing the F6 key. For the most dramatic results, press the F6 key (so that the box moves to the right window), then press the TAB key, which makes that window show columns I through L, as desired. So the right window shows columns I through L, while the left window still shows columns A through D.

If you ever want to move the box back to the left window, press the F6 key again. If you ever want to get rid of the two windows and return to a normal screen, say Worksheet Window Clear. The computer will clear away the two windows and return your screen to normal.

If you say Worksheet Window Horizontal (instead of Worksheet Window Vertical), you'll view the worksheet through two windows separated by a horizontal line. One window will be at the top of the screen, the other will be at the bottom. For the most dramatic effects, move the box to the bottom window (by pressing the F6 key) and then press the PAGE DOWN key (so that the view in that window changes dramatically).

After you give a Worksheet Window Vertical command or a Worksheet Window Horizontal command, you can't give another Worksheet Window command, until you undo the one you gave. To undo it, say Worksheet Window Clear.

MANUAL RECALCULATION

Suppose you've created a gigantic spreadsheet, containing dozens of rows and columns (all filled with numbers, formulas, and words), and you want to change four of the numbers.

The normal way is to point to the cell containing the first number, retype the number, press the ENTER key, and then *wait for the computer to recalculate all the formulas that use the number*. If the spreadsheet is large, hundreds of cells might contain formulas using the number, and you must wait a long time for the computer to recalculate them all.

When the computer has finished recalculating, you point to the second number to change, retype it, press the ENTER key, and again *wait for the computer to relcalculate all the formulas using that number*. Next, retype the third number, press the ENTER key, and *wait*. Then retype the fourth number, press the ENTER key, and *wait*.

So to change the four numbers, you must wait four times for the computer to recompute all relevant formulas in the spreadsheet. You'll be annoyed when you have to wait so long, four times!

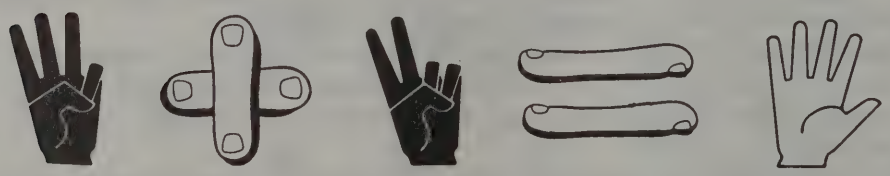
To avoid waiting four times, first say Worksheet Global Recalculation Manual, then move around the spreadsheet and retype the four numbers. When you press the ENTER key after each number, the computer will *not* automatically recalculate all the formulas; instead, the computer will put onto the screen a note saying "CALC", which means "I ought to recalculate all the formulas, but I'm not going to bother." Since the computer doesn't bother to recalculate all the formulas, the computer is immediately ready for you type more numbers; you do *not* have to wait for the computer.

After you've typed all four numbers (without ever having to wait for the computer), and you've typed the ENTER key after each number, and you're finally ready for your coffee break, tap the F9 key. Tapping the F9 key makes the computer recalculate all the formulas. While it's recalculating, go have your cup of coffee. When you come back, all the formulas will have been recalculated, and the "CALC" note will be gone.

Using that method, you must wait for the computer only once (during your coffee break), instead of waiting four times.

That method is called *manual recalculation*. Instead of recalculating automatically each time you press the ENTER key, the computer recalculates when you press the F9 key, which manually triggers the recalculation.

If you ever want the computer to go back to recalculating automatically, say Worksheet Global Recalculation Automatic.



FINAL STEPS

YOUR LAST FIVE STEPS

After you've finished creating your spreadsheet, you'll want to do five things:

Make the columns of numbers look prettier, by aligning their decimal points.
Rearrange the data, to put it in numerical or alphabetical order.
Copy the data onto paper.
Copy the data onto a disk.
Move on to a different spreadsheet or task.

Here's how. . . .

FORMAT THE NUMBERS

Normally, the screen displays numbers like this:

```
        A       B       C       D       E       F       G       H
1     1538.4
2     -0.739
```

If the numbers in your spreadsheet represent money, say Worksheet Global Format Fixed and then press the ENTER key. It makes the spreadsheet show 2 digits after each decimal point and round each amount to the nearest penny, so that the numbers look like this:

```
        A       B       C       D       E       F       G       H
1     1538.40
2       -0.74
```

If you type a comma instead of saying Fixed, the computer will also put commas in large numbers and put parentheses around negative amounts, so that the spreadsheet looks like this:

```
        A       B       C       D       E       F       G       H
1   1,538.40
2     (0.74)
```

If you say Currency instead of Fixed, the computer will also put dollar signs in front of the numbers, like this:

```
        A       B       C       D       E       F       G
1   $1,538.40
2     ($0.74)
```

But before giving that command, you must widen the columns, to allow enough room to insert the dollar signs. To widen the columns, say Worksheet Global Column-width, then type a large number (at least 10), and then press the ENTER key.

If you say Percent instead of Fixed, the computer will express each number as a percentage, by putting a percent sign after the number and multiplying the number by 100, like this:

```
        A       B       C       D       E       F       G
1   153840.00%
2     -73.90%
```

If you change your mind, and want to return to the traditional format (instead of Fixed or comma or Currency or Percent), say Worksheet Global Format General.

All those commands begin by saying Worksheet Global, because they change the formats of *all* the numbers in the worksheet.

Here's how to change the formats of just a *range* of cells. . . .

Point to the beginning of the range.
Say Range Format.
Choose a format: say Fixed or Currency or Percent or General or a comma.
Press the ENTER key (unless you chose General).
Point to the end of the range.
Press the ENTER key.

If you format a range (by saying Range Format), and later give a different format to the worksheet as a whole (by saying Worksheet Global Format), the format given in the Worksheet Global Format command will affect most of the worksheet but won't affect the range you already formatted.

SORT THE DATA

This spreadsheet shows how Sue, Al, and Pedro scored on a test:

```
      A       B       C       D       E       F       G       H
1   Sue        42
2   Al          7
3   Pedro     100
```

Let's rearrange the table, so that the names are in alphabetical order. Here's how. . . .

Say Data Sort. The computer will display this *sort menu*:

```
Data-range  Primary-key  Secondary-key  Reset  Go  Quit
```

From that menu, choose Data-range (by tapping the D key), then type the range of the table, like this:

```
A1.B3
```

Press the ENTER key at the end of that line.

The computer will display the sort menu again. Choose Primary-key (by tapping the P key). Point at the column containing the names you want to alphabetize, then press the ENTER key.

The computer will ask, "Sort order (A or D)?" Tap the A key, then press the ENTER key.

The computer will display the sort menu again. Choose Go (by tapping the G key). The computer will go alphabetize the names and rearrange the entire table accordingly, so that your screen shows this alphabetized table:

```
      A       B       C       D       E       F       G       H
1   Al          7
2   Pedro     100
3   Sue        42
```

To rearrange the table so that the scores are in numerical order, say Data Sort again. The computer will display the sort menu again. Last time, we chose a Data-range of A1 through B3; since the computer still remembers what we chose last time, we don't have to say Data-range again.

Choose Primary-key, point at the column containing the numbers you want to sort, and press the ENTER key. When the computer asks, "Sort order (A or D)?", type an A and press the ENTER key.

The computer will display the sort menu again. Choose Go. The computer will put the scores in numerical order and rearrange the entire table so that it looks like this:

```
      A       B       C       D       E       F       G       H
1   Al          7
2   Sue        42
3   Pedro     100
```

Now the numbers in column B are in *ascending order*. If you want the numbers to be in *descending order* instead, go through the sorting procedure again; but when

the computer asks, "Sort order (A or D)?", type D instead of A. The computer will put the scores into descending order, like this:

```
     A       B       C       D       E       F       G       H
1    Pedro     100
2    Sue        42
3    Al          7
```

PRINT ON PAPER

To print onto paper, first say Print Printer (by tapping the slash key and then twice tapping the P key). The screen will display this *print menu*:

```
Range  Line  Page  Options  Clear  Align  Go  Quit
```

From that menu, choose Range (by tapping the R key), then type the range you want to print. For example, to print from cell A1 to cell H20, type this—

```
A1.H20
```

and press the ENTER key at the end of that line.

The print menu will appear on the screen again. From the print menu, choose Go (by tapping the G key). The printer will go print the range onto paper.

When the printer has finished, the screen will display the print menu again. If you want to print another copy of the same range, choose Go again. If you want to print a different range instead, choose Range, then type the range, then choose Go.

Whenever you're done using the printer, choose Quit from the print menu. That makes the print menu disappear, so that you can do anything else you wish.

SAVE ON DISK

To copy the spreadsheet onto disk B, say File Save. (Exception: if you're using Release 2 instead of 1A, and your 1-2-3 System Disk hasn't been used before, "File Save" won't work well until you obey page 13 of Lotus's own tutorial manual.)

After saying File Save, invent a name for the spreadsheet. For example, if you want the spreadsheet to be named FRED, type FRED and then press the ENTER key.

The computer will try to copy the spreadsheet onto disk B and call the spreadsheet "FRED". The computer will succeed—unless the disk contains a file named FRED already. If the disk already contains a FRED, the computer will display this menu:

```
Cancel  Replace
```

If you choose Replace, the computer will copy the spreadsheet onto the disk, call it FRED, and make it replace the previous file named FRED. If you choose Cancel instead, the computer will cancel your request to copy the spreadsheet onto the disk, so the original disk file named FRED will remain intact.

To see a directory of all the spreadsheets on disk B, say File Retrieve. The computer will display a directory of all the spreadsheets on disk B. If you don't want to use any of those spreadsheets at the moment, tap the BREAK key while holding down the CONTROL key. If you *do* want to use one of those spreadsheets, point to the one you want to use and press the ENTER key: the computer will copy that spreadsheet from the disk to the RAM and show that spreadsheet on your screen.

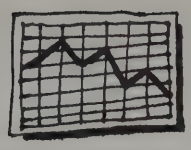
To erase a spreadsheet from disk B, say File Erase Worksheet. The computer will display a directory of all the spreadsheets on disk B. If you change your mind and don't want to erase any of them, tap the BREAK key while holding down the CONTROL key. If you *do* want to erase one of them, point to it, press the ENTER key, and then tap the Y key (to confirm that Yes, you really want to erase it).

QUIT

When you've finished using 1-2-3, say Quit Yes, by typing /QY. When you see the next menu, type E (for Exit). If you're using Release 1A, then type Y.

The computer will say:

```
A>
```



GRAPHS

SIMPLE GRAPHS

Suppose you're running a company that sells Day-Glo Pink Hair Dye. (Your motto is: "To brighten your day, stay in the pink!") You have two salespeople, Joe and Sue.

Joe has worked for you a long time, and sells about $8,000 worth of dye each month. Sue joined your company recently and is rapidly getting better at encouraging people to turn their hair pink.

This spreadsheet shows how many dollars worth of dye Joe and Sue sold each month:

	A	B	C	D	E	F	G	H
1		January	February	March	April			
2	Joe	8000	6500	8200	7400			
3	Sue	2000	4300	6800	12500			

The spreadsheet shows that Joe sold about $8000 worth of dye each month: the exact amount varied from $6500 to $8200. Sue is a trainee who sold just $2000 worth in January but whose monthly sales have zoomed up to $12500 in April.

Let's turn that spreadsheet into a graph. First, type the spreadsheet. Then say Graph. The computer will display this *graph menu*:

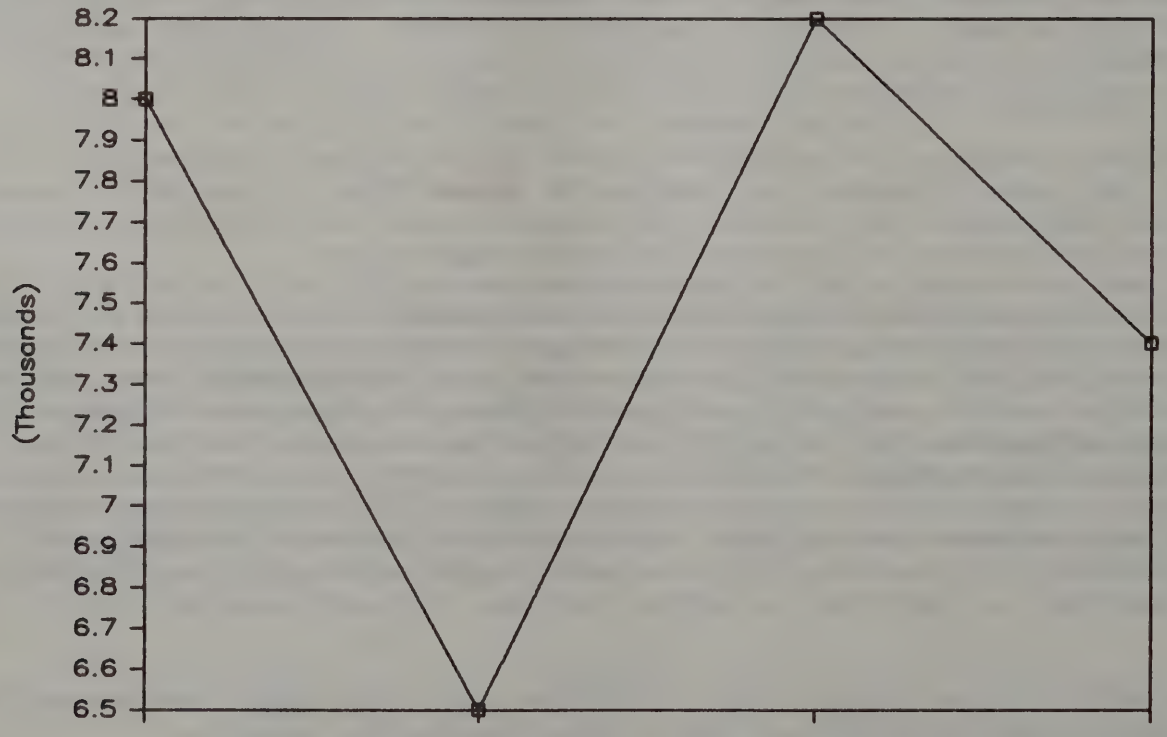
```
Type  X  A  B  C  D  E  F  Reset  View  Save  Options  Name  Quit
```

Choose A from the menu. The computer will say, "Enter first data range." Tell the computer which range contains Joe's data; to do that, type—

```
B2.E2
```

and press the ENTER key.

The computer will display the graph menu again. If your computer contains an IBM color graphics card (or a clone), choose View from the menu; it makes the screen display this graph:



Notice that the computer automatically scales the graph, so that the graph fits on the screen.

When you've finished admiring the graph, press the ENTER key, which makes the computer display the graph menu again.

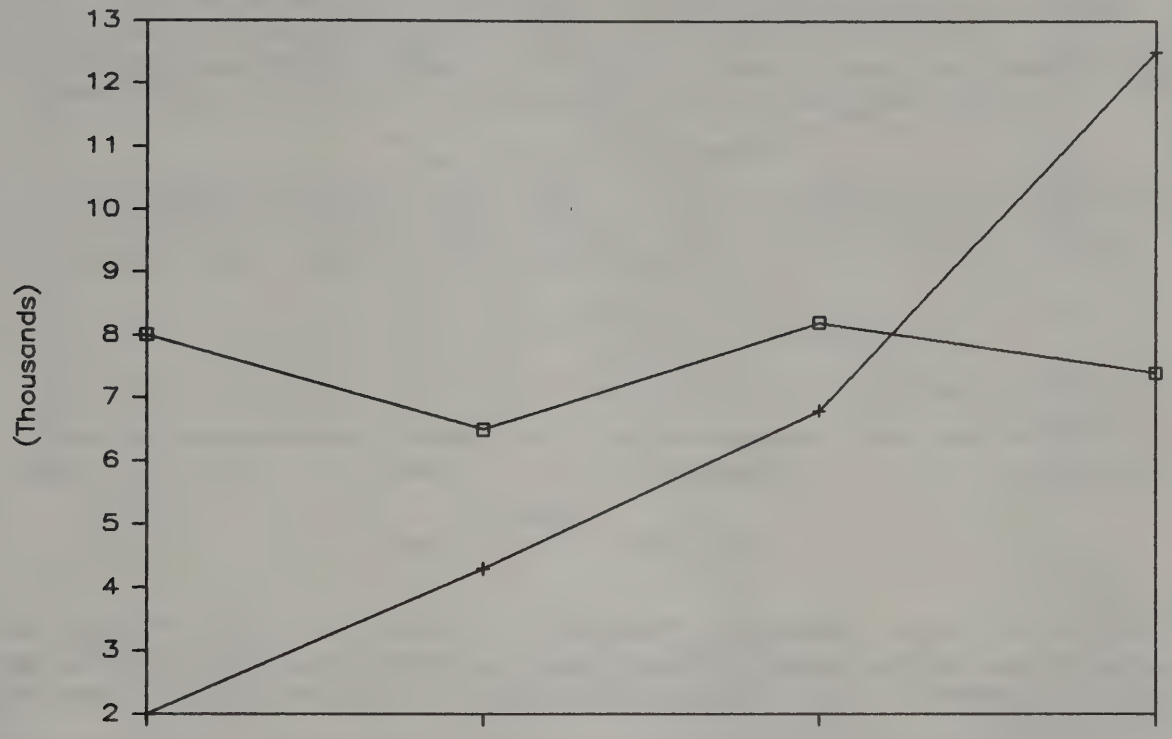
(If your computer does *not* contain a graphics card, you can't see graphs on your screen, and so the View command doesn't work; but I'll explain later how to print the graphs onto paper.)

MULTIPLE GRAPHS

That graph shows Joe's sales. Let's make it show Sue's also.

Joe is called "person A"; Sue is called "person B". To make the graph include Sue's sales, choose B from the graph menu, then tell the computer Sue's range, by typing "B3.E3" and pressing the ENTER key.

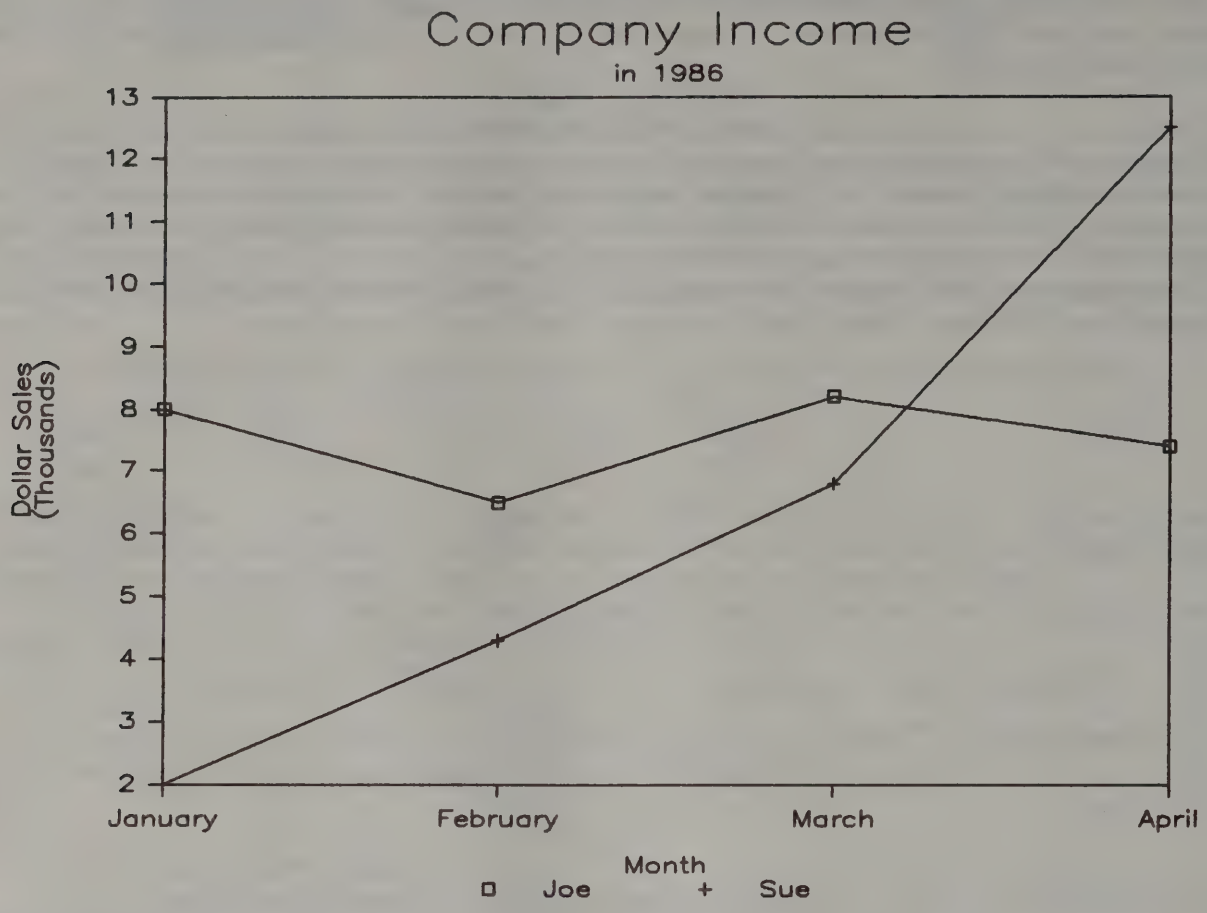
The computer will display the graph menu again. To see the new graph (which includes Sue), choose View from the menu. You'll see this graph:



To create that graph, we defined Joe's data by choosing A from the graph menu, and we defined Sue's data by choosing B. The 1-2-3 program lets you graph up to six people, by calling them A, B, C, D, E, and F, and choosing those letters from the graph menu.

LABELING THE GRAPHS

Let's label the graph we created, so that it looks like this:



Here's how. . . .

To write the months "January", "February", "March", and "April" along the X axis at the bottom of the graph, choose X from the graph menu, then type the range "B1.E1" (because that's where those months occur in the spreadsheet), then press the ENTER key.

To write the other labels ("Company Income", "Dollar Sales", "Month", "Joe", and "Sue"), choose Options from the graph menu. The computer will display this *options menu*:

```
Legend  Format  Titles  Grid  Scale  Color  B&w  Data-labels  Quit
```

To make "Company Income" be the first line of the graph, choose Titles from the options menu, then type F (which stands for First), then type—

```
Company Income
```

and press the ENTER key. The computer will display the options menu again.

To make "in 1986" be the second line of the graph, choose Titles from the options menu, then type S (which stands for Second), then type—

```
in 1986
```

and press the ENTER key. The computer will display the options menu again.

To make the word "Month" appear below the X axis, choose Titles from the options menu, then type X, then type—

```
Month
```

and press the ENTER key. The computer will display the options menu again.

To write the phrase "Dollar Sales" next to the Y axis, choose Titles from the options menu, then type Y, then type—

```
Dollar Sales
```

and press the ENTER key. The computer will display the options menu again.

To make the bottom of the graph show a legend saying that the box represents Joe, choose Legend from the options menu, then type A, then type Joe and press ENTER. To write a legend saying that the plus sign represents Sue, choose Legend from the options menu, then type B, then type Sue and press ENTER.

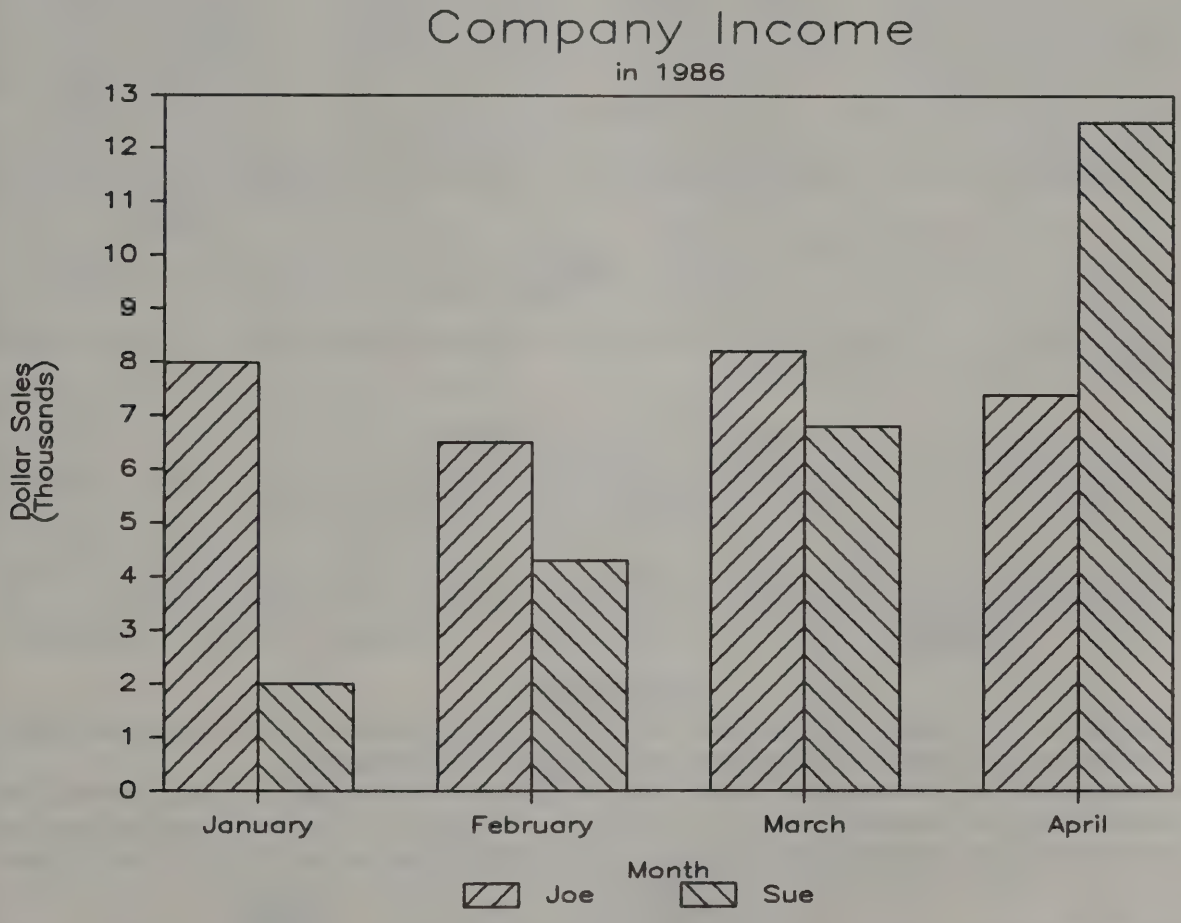
If you have a color monitor, choose Color from the options menu. Then all information about Joe will appear in magenta (reddish purple), and all information about Sue will appear in cyan (greenish blue)—unless you're using Release 1A, which chooses different colors instead.

When you're done defining those options, choose Quit from the option menu. That makes the computer switch from the option menu to the graph menu.

To see the final graph, including all those options, choose View from the graph menu. Press the ENTER key when you've finished viewing the graph.

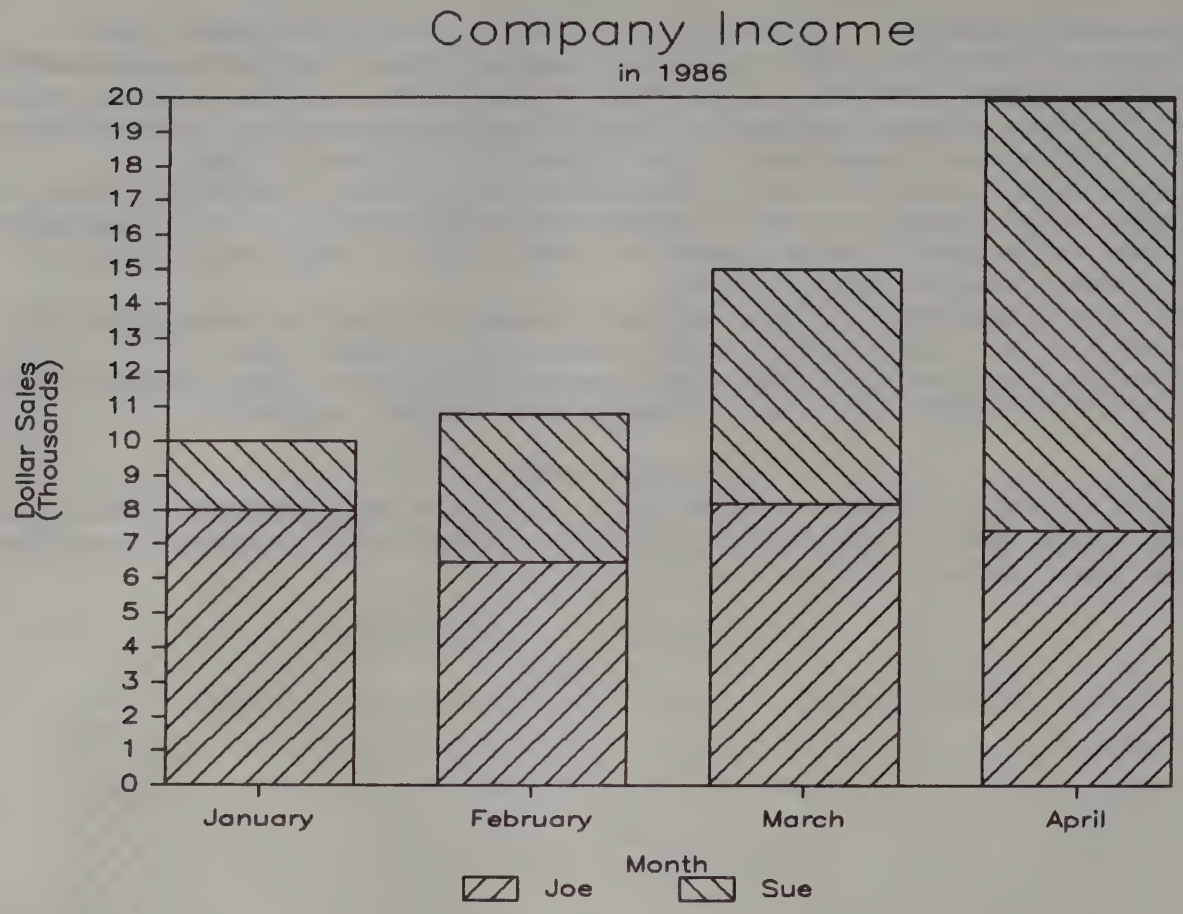
BAR GRAPHS

That style of graph is called a *line graph*. To see a bar graph instead, choose Type from the graph menu, then press B, which stands for Bar. When you choose View from the graph menu, you'll see this bar graph:



That's what you'll see if your monitor is monochrome. If your monitor can handle colors, and you chose the Color option with Release 2, Joe's bars will be made of solid magenta, and Sue's bars will be made of solid cyan, instead of diagonal stripes.

To see a *stacked* bar graph, choose Type from the graph menu, then press S, which stands for Stacked. When you choose View from the graph menu, you'll see this stacked bar graph:



If you ever want to return to a line graph, choose Type from the graph menu, then press L, which stands for Line.

OTHER TYPES OF GRAPHS

Besides choosing B (for bar graphs), S (for stacked-bar graphs), and L (for line graphs), you can choose P (for pie charts) and X (for X-Y plots).

If you ask for a pie chart, the computer will draw the chart for just one person: the computer will draw a pie chart for person A (Joe) but not for person B (Sue).

If you choose an X-Y plot, the X range must consist of numbers (such as -2.3, .7, 4.9, and 17) instead of words (such as January, February, March, and April).

FINISHING

When you finish graphing, choose Quit from the graph menu. That makes the graph menu disappear, so that you can give other kinds of commands.

The computer will remember what kind of graph you wanted. If you revise the numbers in the spreadsheet, you can see the revised graph by just tapping the F10 key. The graph will immediately appear. After you've admired it, tap the F10 key again.

PRINTING GRAPHS ONTO PAPER

To print a graph onto paper, you must go through several steps.

First, prepare the graph as if you were going to View it.

Next, save the graph onto a disk. To do that, choose Save from the graph menu, then type a name for the graph (invent whatever name you like, such as SALES), then press the ENTER key. The computer will copy the graph onto disk B.

The computer will display the graph menu again. Choose Quit. The graph menu will disappear.

Then say Quit Yes. When you see the next menu, type P (for Print-graph).

The computer will say:

```
Insert Print Graph Disk in drive A.
Press RETURN to continue.
```

Obey the computer: remove the 1-2-3 System Disk from drive A, put the Print Graph Disk into drive A, and then press the ENTER key.

The computer will display this *print graph menu*:

```
Image-select  Settings  Go  Align  Page  Exit
```

(Release 1A's menu is slightly different.)

If your Print Graph Disk has never been used before, you must *configure* it. (To configure using Release 2, say Settings Hardware Graphs-directory, then press the B key, then the colon key, then the ENTER key, then say Printer, move the box to the kind of printing system you want to use, press the ENTER key, then say Quit Save. To configure using Release 1A, say Configure, then follow the instructions in Lotus's manual.)

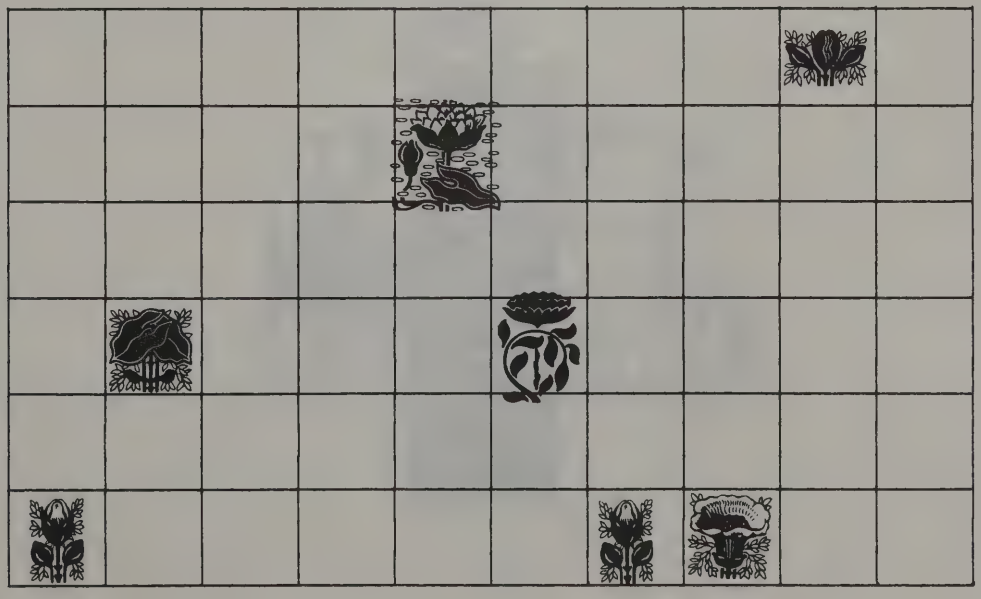
From the print graph menu, choose Image-select. (Release 1A calls it Select.) The screen will display a directory of all the graphs on disk B. Move the box to the name of the graph you want to print, then press the SPACE bar. A number sign will appear in front of the graph's name. Press the ENTER key.

From the print graph menu, choose Go. The computer will print the graph.

From the print graph menu, choose Exit. (Release 1A calls it Quit.) The computer will ask whether you're sure; say Yes. Then say 1 (for 1-2-3). The computer will say:

```
Insert 1-2-3 System Disk in drive A.
Press RETURN to continue.
```

Obey the computer: remove the Print Graph Disk from drive A, put the 1-2-3 System Disk into drive A, and then press the ENTER key.



WOW! YOU FINISHED CHAPTER 8: SPREADSHEETS!
TURN THE PAGE FOR CHAPTER 9: DATA-MANAGEMENT SYSTEMS.

CONCEPTS

WHAT'S A DATA-MANAGEMENT SYSTEM?

A *data-management system* is a program that manipulates lists of facts. For example, it can manipulate a mailing list, a phone directory, and any other lists you wish, concerning your customers, suppliers, employees, friends, enemies, students, teachers, clients, libraries, and hobbies; it puts all that data about your business and life onto the disk, and then lets you retrieve that data easily. It turns your computer into an electronic filing cabinet.

It's called a *data-management system* or *information-retrieval system*; the terms are synonymous.

FILE-CABINET JARGON

In an old-fashioned office that lacks a computer, you'll see a filing cabinet containing several drawers. One drawer's labeled CUSTOMERS; another drawer's labeled EMPLOYEES; another drawer's labeled SUPPLIERS. Each drawer contains many index cards, alphabetized.

For example, the drawer labeled CUSTOMERS contains a card about each customer; the first card might be labeled "ADAMS, JOAN"; the last card might be labeled "ZABRONSKI, JILL". The first card contains all known information about Joan Adams: it contains her name, address, phone number, a list of everything she's purchased from the office, how much she paid, how much she still owes, and other personal information about her. That card is called her *record*. Each item of information on that card is called a *field*.

If the card is a pre-printed form, it allows a certain amount of space for each item: for example, it might allow only 30 characters for the person's name. The number of characters allowed for a field is called the *field's width*. In that example, the width of the NAME field is 30 characters.

Each drawer is called a *file*. For example, the drawer that contains information about customers is called the *customer file*; another drawer is the *employee file*; another drawer is the *supplier file*.

The entire filing cabinet—which contains all the information about your company—is called the *data base*.

A SAMPLE FILE

Here's a file, about amazing students in the School of Life:

Last name: Smith *First name: Suzy*
Age: 4 *Class: 12*
Comments: Even though she's only 4 years old, she's already in the twelfth grade and will graduate this year, because she's fast.

Last name: Bell *First name: Clara*
Age: 21 *Class: 10*
Comments: The class clown, she never quite made it through high school. She had lots of fun trying, though. Too bad she's as slow as a hibernating bear.

Last name: Smith *First name: Buffalo Bob*
Age: 7 *Class: 2*
Comments: Boringly normal, he's jealous of his sister Suzy.

Last name: Kosinski *First name: Stanislaw*
Age: 16 *Class: 11*
Comments: He dislikes Polish jokes.

Last name: Ketchopf *First name: Heinz*
Age: 57 *Class: 1*
Comments: His pour grades make him the slowest Ketchopf in the West.

Last name: Nixon *First name: Tricky Dick*
Age: 73 *Class: 13*
Comments: The unlucky President, he disappointed our country.

Last name: Walter *First name: Russy-poo*
Age: 38 *Class: 0*
Comments: This guy has no class.

That file consists of seven records: Suzy Smith's record, Clara Bell's record, Buffalo Bob Smith's record, Stanislaw Kosinski's record, Heinz Ketchopf's record, Tricky Dick Nixon's record, and Russy-poo Walter's record.

Each record consists of five fields: last name, first name, age, class, and comments. The age and class fields are narrow; the comments field is very wide.

DATA MANAGEMENT VERSUS WORD PROCESSING

A word-processing system lets you write a memo, put the memo onto a disk, edit the memo, and copy the memo onto a printer. A data-management system is quite similar. Like a word-processing system, a data-management system lets you write information, put the information onto a disk, edit the information, and copy the information onto a printer.

In a word-processing system, the information's called a *document*, which consists of paragraphs, which in turn consist of sentences. In a *data-management* system, the information's called a *file* (instead of a document); it consists of records, which in turn consist of fields.

Because a data-management system is so similar to a word-processing system, a word-processing system can act as a crummy data-management system. But a *good* data-management system offers the following "extras", which the typical word-processing system lacks. . . .

A good data-management system can alphabetize.

A good data-management system can automatically rearrange information, to put it in numerical order.

A good data-management system can check for criteria. For example, you can tell a good data-management system to check which customers are women under 18 who have light red hair and live in a red-light district. You can even make the data-management system print their names and addresses on mailing labels, in zip code order. You can even make the data-management system create your own phone book, containing the names and phone numbers of just those women. As you can see, data-management systems are very potent, and can be nasty tools for invading people's privacy!

PFS FILE

WHAT'S PFS FILE?

The most popular data-management system is *PFS File*, because it's very easy to understand and performs a wide variety of tasks.

It comes on a disk. It's published by *Software Publishing Company*, which also publishes the PFS Write word-processing program.

Versions. The fanciest version of PFS File runs on the IBM PC. It's called *version B*. It lists for $140; discount dealers charge $75. It runs even on the PC Junior and clones.

Older, inferior versions for the IBM PC are called *version A* and *Filing Assistant*. Version A is no longer sold. IBM still sells Filing Assistant for a theoretical price of $149, but often gives it away for free when you buy an IBM computer.

Other versions of PFS File are available for the Apple 2-plus & 2e & 2c & Macintosh, the Commodore 64, and the Radio Shack Models 3 & 4 & 4P. Those versions are slightly worse than the best IBM version (which is version B), and therefore cost less.

Capabilities. PFS File can handle long files. Each file can contain up to 4 megabytes, and up to 32,000 records. Each record can contain up to 3,200 fields. But since the entire file must fit on a single disk, the only way to handle such a long file is to buy a hard disk.

After buying PFS File, you'll also want to buy *PFS Report*, which prints fancy reports analyzing the files created by PFS File.

Buy both PFS File and PFS Report, because that pair of programs is good enough to handle most files for most businesses.

HOW TO BUILD A FILE

Here's how to build a file by using the best version of PFS File, which is version B on the IBM PC.

Getting to the main menu. First, make the computer say "A>". (To do that, put a system disk into drive A, then turn on the computer, then type the date and time.)

Remove the system disk from drive A, insert the PFS File disk instead, and type the word "file", so your screen looks like this:

```
A>file
```

When you press the ENTER key at the end of that command, the computer will copy the entire PFS File program from the disk to the RAM and then start running the program. The screen will display this *main menu*:

```
PFS FILE MAIN MENU

1.  Design file
2.  Add forms
3.  Copy
4.  Search/update
5.  Print
6.  Remove
E.  Exit
```

Inserting blank disks. Remove the PFS File disk from drive A. Then into both drives (A and B), insert *formatted* disks that are blank or nearly blank, so that they have lots of room left on them.

PFS File uses drive B very rarely: just to make copies and to sort. So if you're not going to tell PFS File to make copies or to sort, you don't have to bother putting a disk into drive B.

Designing the file. Let's design a file called STUDENTS, about the students in the School of Life. To do that, look at the main menu, from that menu choose "1. Design file", and also tell the computer that you want the file to be called STUDENTS. To do all that, tap the 1 key, then the TAB key (which is next to the Q key), then type STUDENTS, then press the ENTER key.

Then the screen will display a different menu. From that menu, choose "1. Create a new file". To do that, tap the 1 key and then the ENTER key.

Most of the screen will become blank. On the blank screen, type a form, which you'll fill in later. For example, if you want to store each student's last name, first name, age, and class, plus comments, type this form:

```
Last name:                              First name:
Age:                Class:
Comments:
```

Notice you should type the word "Last", then a space, then the word "name", then a colon, then several spaces, then the phrase "First name", then a colon, then the ENTER key, etc. That form creates five fields: last name, first name, age, class, and comments. When you've finished typing the form, copy it onto disk A, by pressing the F10 key.

Then the computer will display the main menu again.

Adding forms. Although the computer has put a file called STUDENTS onto the disk, and although that file is organized so that each record will consist of five fields, the file doesn't contain any records yet; it doesn't contain the names and data about any specific students yet.

To add the names of specific students and the data about them, look at the main menu and choose "2. Add forms". To do that, tap the 2 key and then the ENTER key.

The screen will show the form you designed:

```
Last name:                              First name:
Age:                Class:
Comments:
```

Then fill in the blanks. For example, if one of the students is Suzy Smith, fill in the blanks like this:

```
Last name: Smith                        First name: Suzy
Age: 4              Class: 12
Comments: Even though she's only 4 years old, she's already in the twelfth grade
and will graduate this year, because she's fast.
```

To fill in the blanks that way, type Smith, then press the TAB key (which moves the cursor to the next field), then type Suzy, then press the TAB key, then type 4, then press the TAB key, then type 12, then press the TAB key, then type the commentary sentence.

When you've finished filling in Suzy Smith's form, copy it onto disk A, by pressing the F10 key.

Then the computer will show a blank form again:

```
Last name:                              First name:
Age:                Class:
Comments:
```

Fill in the blanks again, for the next student (Clara Bell). Then copy that student's record onto the disk, by pressing the F10 key.

Repeat that process for each student. When you've finished the last student, press the F10 key, to copy that student's data onto the disk. Then the computer will show a blank form again (in case you want to add another student), but cancel that blank form, by pressing the ESCAPE key (which says Esc on it).

The computer will display the main menu again.

HOW TO SEARCH

To search through the file, to find a particular student, look at the main menu and choose "4. Search/update".

The screen will show again the form you designed:

```
Last name:                              First name:
Age:              Class:
Comments:
```

If you fill in a few of the blanks, the computer will fill in the rest. For example, if you fill in just the last name (Smith) and then press the F10 key, the computer will find everybody whose last name is Smith, and will show you each Smith's record, one at a time.

Whenever the computer finds a Smith's record, it pauses, to let you look at the record. When you've finished looking, press the F10 key again, which makes the computer find the next Smith.

While you're looking at a Smith's record, you can revise the record. (Revising is also called *editing* or *updating*.) To revise the record you're looking at, move the cursor to the field you want to revise, by tapping the TAB key several times. Then retype that field.

After the computer finds all the Smiths and displays their records on the screen, the computer counts how many Smiths it found, and displays that number on your screen.

Then press the ENTER key. The computer will display the main menu again.

Fancy searches. If you tell the computer to search for "S.." instead of "Smith" (by putting "S.." in the last name field), the computer will find everybody whose last name begins with "S". If you say "..th", the computer will find everybody whose last name ends with "th". If you say "..m..", the computer will find everybody whose name contains an "m". If you say "S????", the computer will find everybody whose name contains an S followed by exactly 4 more letters.

If you say "/Smith", the computer will find everybody whose name is *not* Smith. If you say "/..m..", the computer will find everybody whose name does *not* contain an m.

Suppose you want to find Kosinski's record, but you can't remember how he spells his name. You don't remember whether it's "Kosinski" or "Cosinski" or "Kozinski" or "Kosinscki" or "Kosinsky"; you remember merely that it includes the letters "in". If you put "..in.." into the last name field, the computer will find everybody whose name contains the word "in"—and it will find Kosinski's record.

If you fill in just the age, the computer will find everybody who has that age. To find everybody who's 10 years old, put "=10" in the age field. To find everybody who's less than 10, say "<10". To find everybody who's greater than 10, say ">10". To find everybody who's between 10 and 18, say "=10..18". To find everybody who's *not* between 10 and 18, say "/=10..18". To find everybody whose age was left blank, say "/..". To find everybody whose age was filled in, say "..".

If you fill in just the comment, the computer will find everybody who has that comment. For example, if you put "..slow.." into the comment field, the computer will find everybody who's comment mentions "slow": it will find Clara Bell (who's "as slow as a hibernating bear") and will also find Heinz Ketchopf (who's "the slowest Ketchopf in the West").

Suppose you want to find everybody who's old and stupid. Specifically, suppose you want to find everybody whose age is over 20 and who still hasn't graduated from the eighth grade. In the age field put ">20", and in the class field put "<9". The computer will find Heinz Ketchopf (who's 57 years old and in the first grade) and Russy-poo Walter (who's 38 years old and whose class is 0).

If you leave all the fields blank, so that you're not telling the computer to search for anything in particular, the computer will show you *all* the records in the file, without any restrictions.

When you're dealing with forms (inventing them, filling them in, or editing them), you can move the cursor easily.

By tapping the four arrow keys, you can move the cursor up, down, left, or right.

To move the cursor ahead to the next field, tap the TAB key. To move the cursor back to the previous field, tap the TAB key while holding down the SHIFT key.

To move the cursor down, to the beginning of the line underneath, tap the ENTER key. To move the cursor up, to the upper-left corner of the screen, tap the HOME key.

If you want to create a form that's too long to fit on the screen, put the first part of the form on the screen, then tap the PAGE DOWN key (which says PgDn), then type the next screenful. After typing the second screenful, if you still need more space, tap the PAGE DOWN key again, and type another screenful. To return to the previous page of the form, tap the PAGE UP key.

If you want to write a long comment about an unusual student, and the comment's too long to fit on the screen, type as much of the comment as will fit, then press the PAGE DOWN key, then type the rest of the comment.

Erasing your mistakes. To erase the character just left of the cursor, press the BACKSPACE key. To erase the character where the cursor is, press the DELETE key (which says Del on it).

If you accidentally choose the wrong number from a menu, or give some other wrong command, press the ESCAPE key. The computer will cancel the command and display the main menu again.

Function keys. On the left side of the keyboard, you'll see ten function keys, numbered from F1 to F10. Each of them has a name.

F1: HELP	*F2: PRINT*
F3: REMOVE	*F4: ERASE*
F5: DATE	*F6: TIME*
F7: CALCULATE	*F8: DITTO*
F9: NULL	*F10: CONTINUE*

If you tap the HELP key (F1), the screen will show a message that reminds you how to use PFS File and what to type next. After you've read the message, make it disappear, by tapping the ENTER key.

When you turned the computer on, you told it the date and time. The computer remembers the date and time and continually keeps track of how they change, by using the computer's built-in clock. Whenever you want to type the date, tap the DATE key (F5): it makes the computer put today's date onto the screen, where the cursor is. To type the time, tap the TIME key (F6).

If a person's record is on the screen, and you want to copy it onto paper, tap the PRINT key (F2), then tap the ENTER key. The printer will print the person's entire record onto paper.

If you tap the ERASE key (F4), the computer will erase all the data on the screen. (It will not erase the names of the fields.) Then change the data, by retyping it however you wish.

Suppose a person died, and you want to remove the person's record from the file. First, get the person's record onto the screen (by searching for it); then tap the REMOVE key (F3); then tap the ENTER key (to confirm that you really want to destroy the person's record). The computer will delete that person's record from the disk.

Suppose you're adding Alice to the file, and her age is the same as the previous person's. Move the cursor to the age field; but instead of typing her age, just tap the DITTO key (F8). The computer will automatically type her age, and make it the same as the previous person's.

So when you're adding a new record, if you move the cursor to a field and then tap the DITTO key, the computer will make that field become a duplicate of what you typed for the previous record.

The NULL key (F9) doesn't do anything yet. Maybe the next version of PFS File will make it do something.

When you've finished manipulating a form on the screen, tap the CONTINUE key (F10), to move on to the next form or the next activity.

The CALCULATE key (F7) is special. Here's how to use it. . . .

CALCULATED FIELDS

Suppose you run a diet clinic, and one of your clients is Martha Magdalene (Mary's sister). She used to weigh 167 pounds; but now, thanks to your clinic, she weighs just 110.

Let's make the computer print this report:

```
Last name: Magdalene                    First name: Martha

Weight before: 167
Weight after : 110

Weight lost    : 57
```

Let's make the computer print that kind of report about *each* of your clients. Let's make the computer automatically compute each client's "Weight lost", by subtracting the "Weight after" from the "Weight before". And let's make the computer put all the data about all the clients onto a disk, so that you can analyze the data further whenever you wish.

To do all that, design this form by using PFS File:

```
Last name:                              First name:

Weight before:
Weight after :

Weight lost    :
```

The next step is to teach the computer that "Weight lost" means: "Weight before" minus "Weight after". Here's how. . . .

Go back to the PFS File main menu, and again choose "1. Design file". When the next menu appears on the screen, choose "4. Specify formulas for calculated items".

The form you designed will appear on the screen again. Notice that the bottom three fields represent numbers. In the first of those numeric fields, write "#1"; in the next numeric field, write "#2"; in the next numeric field, write "#3", like this:

```
Last name:                              First name:

Weight before: #1
Weight after : #2

Weight lost    : #3
```

We want the "Weight lost" (which is #3) to be the "Weight before" (which is #1) minus the "Weight after" (which is #2). So the "Weight lost" is defined by this formula: #3 = #1 - #2. Put that entire formula in the "Weight lost" field, so that the form finally looks like this:

```
Last name:                              First name:

Weight before: #1
Weight after : #2

Weight lost    : #3 = #1 - #2
```

When you've finished typing that, press the F10 key. The computer will display the main menu again.

Now the file acts normally. For example, to add your clients' names and weights to the file, choose "2. Add forms" from the main menu. The screen will show this form:

```
Last name:                          First name:

Weight before:
Weight after :

Weight lost   :
```

For each client, fill in the form, by typing the client's last name, first name, weight before, and weight after. But do *not* type the weight lost: instead, just tap the CALCULATE key (F7). When you tap that key, the computer will automatically calculate the weight lost (by using the formula #3=#1-#2) and automatically type the answer for you, in the weight lost field.

After the computer has typed the weight lost, do whatever you wish. For example, to print that form onto paper, tap the PRINT key (F2) and then the ENTER key. To copy that form onto disk and continue on to the next form or activity, tap the F10 key.

Receipts. Suppose you run a shop called "Think & Drink", where you sell computers and soft drinks to kids. When Sammy Jones comes into your store and buys two printer ribbons ($8.50 each), a Commodore 64 ($139.50), and two cans of Pepsi (50¢ each), let's make the computer print this receipt:

```
Last name: Jones              First name: Sammy

What bought: printer ribbons  How many: 2    Each:   8.50  Total $:  17.00
What bought: Commodore 64     How many: 1    Each: 139.50  Total $: 139.50
What bought: cans of Pepsi    How many: 2    Each:    .50  Total $:   1.00
-------------------------------------------------------------------------
                                                 Grand total $: 157.50
                                               5% Sales tax $:   7.88
-------------------------------------------------------------------------
                                                 AMOUNT DUE $: 165.38
```

Let's make the computer print receipts for *all* the customers, and put all the data from all the customers onto a disk, so that you can analyze all your store's data whenever you wish.

To do all that, first design this form:

```
Last name:                    First name:

What bought:                  How many:      Each:          Total $:
What bought:                  How many:      Each:          Total $:
What bought:                  How many:      Each:          Total $:
-------------------------------------------------------------------------
                                                 Grand total $:
                                               5% Sales tax $:
-------------------------------------------------------------------------
                                                 AMOUNT DUE $:
```

In that form, the dashed lines are optional: they just make the form look prettier.

After creating that form (including the dashed lines), go back to the PFS File main menu, choose "1. Design file" again, and from the next menu choose "4. Specify formulas for calculated items". The screen will show the form you designed.

Notice which of the fields represent numbers. In the first numeric field, say "#1"; in the second numeric field, say "#2"; in the third numeric field, say "#3" and also say that it equals #1 times #2. Fill in all the numeric fields, so that the form finally looks like this:

```
Last name:                    First name:

What bought:                  How many: #1   Each: #2      Total $: #3=#1*#2
What bought:                  How many: #4   Each: #5      Total $: #6=#4*#5
What bought:                  How many: #7   Each: #8      Total $: #9=#7*#8
-------------------------------------------------------------------------------
                                                    Grand total $: #10=#3+#6+#9
                                                  5% Sales tax $: #11=#10*.05
-------------------------------------------------------------------------------
                                                    AMOUNT DUE $: #12=#10+#11
```

When you've finished typing those formulas, press the F10 key. You'll see the main menu.

To print receipts, choose "2. Add forms". You'll see the form you designed. Fill in the customer's last name and first name, and also fill in the fields marked "What bought", "How many", and "Each". When you type dollar amounts in the field marked "Each", make sure you put two digits after the decimal point: if an item costs $5, make sure you type "5.00" and not just "5".

Do *not* bother filling in the fields marked "Total $", "Grand total $", "5% Sales tax", and "AMOUNT DUE": instead, just tap the CALCULATE key (F7) once. When you tap that key, the computer will automatically calculate and fill in all the fields defined by formulas.

Moreover, the computer will mimic your style. Since you typed two digits after the decimal point in the "Each" field, the computer will compute and print two digits after the decimal point in the totals, grand total, sales tax, and amount due. The computer will even round the sales tax to the nearest penny, and print the rounded amount.

Then press the PRINT key (F2), which prints a receipt on paper, to hand to the customer. Finally, press the F10 key, to copy the receipt onto the disk.

If you follow that procedure, you'll face only one disappointment: the computer is too stupid to know how to line up the decimal points and make the columns neat. If you insist on lining up the decimal points, you must do it yourself. Here's how. . . .

After you press the CALCULATE key and the computer calculates the formulas, press the INSERT key (which says Ins on it). Then move the cursor in front of each number that's misaligned, and tap the space bar several times. That inserts extra spaces before the number, so that the number moves toward the right. Keep tapping the space bar, until the number is far enough to the right so that its decimal point lines up with the other numbers in the column.

When you've finished lining up the decimal points of all the numbers, press the INSERT key again.

MASS REMOVAL

To remove all the Smiths from a file, you could search for them, wait for each of their records to appear on the screen, and then tap the REMOVE key (F3) each time. But tapping the REMOVE key several times is tedious. Here's a short-cut way to remove all the Smiths. . . .

From the main menu, choose "6. Remove". Then the computer will show the form you designed. Indicate which records you want to remove: for example, to remove all the Smiths, type "Smith" in the last name field, and then press the F10 key.

Next, press the ENTER key, to confirm that you really want to remove all the Smiths.

The computer will delete all the Smiths' records from the disk file. It will also count them, and tell you how many Smiths were deleted.

After you admire the count, press the ENTER key. The computer will display the main menu again.

Suppose you've created this file of recent Presidents:

Last name: Nixon *First name: Dick*
Year inaugurated: 1969 *Party: R*

Last name: Ford *First name: Gerry*
Year inaugurated: 1974 *Party: R*

Last name: Carter *First name: Jimmy*
Year inaugurated: 1977 *Party: D*

Last name: Reagan *First name: Ronnie*
Year inaugurated: 1981 *Party: R*

Let's make the computer print, onto paper, information about each of the Republicans.

To do that, you could search for the Republicans, wait for each of their records to appear on the screen, and then tap the PRINT key (F2) each time. But tapping the PRINT key several times is tedious. Here's a more sophisticated way to print all the Republicans onto paper. . . .

From the main menu, choose "5. Print". When the next menu appears on the screen, choose "1. Print forms".

Then the computer will show the form you designed. Indicate which records you want to print on paper: for example, to print all the Republicans, type "R" in the party field, then press the F10 key.

Then the screen will show these *print options*:

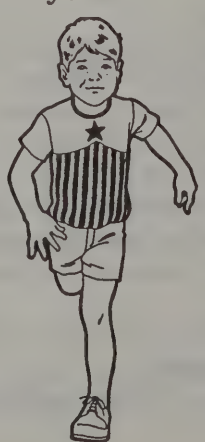
```
PRINT OPTIONS

Pre-defined print spec:
Print item name (Y/N): Y
Print to: PRN:
Number of copies: 1
Lines per page: 66
Pause between pages (Y/N): N
Printer control codes:
```

Attach the printer to the computer, put paper in the printer, and turn the printer on. Press the ENTER key. The screen will display the form you designed. Type an "x" in each field that you want printed onto paper. For example, if you want to print just the President's last name and first name, put an x in just those fields, so the screen looks like this:

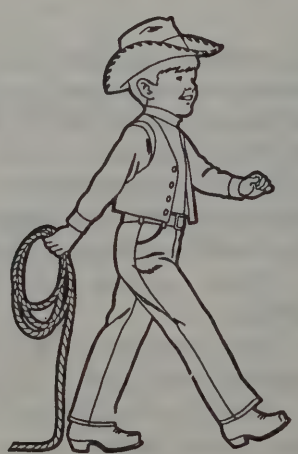
```
Last name: x                    First name: x
Year inaugurated:               Party
```

Then press the F10 key.

Tricky Dicky


Very Gerry

Gimme Jimmy!


Round-em-up, Ronnie!

The printer will do what you requested: it will print just the last name and first name of the three Republicans. On one sheet of paper, it will print:

```
Last name: Reagan
First name: Ronnie
```

On another sheet of paper, it will print:

```
Last name: Ford
First name: Gerry
```

On a third sheet, it will print:

```
Last name: Nixon
First name: Dick
```

Notice it begins by printing the most recent President (Reagan), and ends by printing the most obsolete (Nixon). That's because it thinks you're most interested in the most recent data and want to see that data immediately, before bothering with the older stuff.

If you'd rather have them printed in a different order, say so, by putting an "s" in the field that you want the computer to use for sorting. For example, if you want to see the Presidents in alphabetical order, according to last name, put an "s" in the last name field, like this:

```
Last name: xs                          First name: x
Year inaugurated:                      Party:
```

That makes the computer print Ford's record, then Nixon's, then Reagan's. If you'd rather see the Presidents in the order they were inaugurated, put the "s" in the "Year inaugurated" field instead.

Warning: if you put an "s" in a field that involves numbers, the computer will sort correctly only if each person's number has the same length. For example, if you're running a diet clinic for adults and try to sort your clients according to their weights, the computer will sort correctly only if each person's weight consists of three digits; a person whose weight is less than 100 will be sorted incorrectly.

Normally, the computer prints each field on a separate line, like this:

```
Last name: Reagan
First name: Ronnie
```

If you prefer, you can make the computer print several fields on the same line, like this:

```
Last name: Reagan  First name: Ronnie
```

To do so, put a "+" in the "Last name" field, instead of an "x". The "+" tells the printer to print the field and then stay on the same line, instead of dropping to the line below.

When the printer finishes printing the records of the 3 Republican Presidents, the screen says:

```
Forms printed: 3
```

Press the ENTER key. The screen will display the main menu again.

Changing the print options. If you don't like the way the computer normally prints on paper, you can make it print abnormally instead.

When the screen shows the print options, you can move the cursor around the screen and change the options. For example, if you want to print *two* copies of each form, move the cursor to "Number of copies: 1", and change the 1 to 2. If you want the forms printed in triplicate, change the 1 to 3: that makes the computer print each form 3 times.

If your printer requires you to insert each sheet of paper by hand, make the printer pause at the bottom of each sheet, so that you have a chance to insert the new sheet. To do that, move the cursor to "Pause between pages (Y/N): N", and change the last N to Y.

A normal sheet of paper contains 66 lines of printing. If you want to print on mailing labels instead, move the cursor to "Lines per page: 66", and change the 66 to 6—because the typical mailing label is just large enough to hold 6 lines of printing.

If you want to save paper, you can make the computer print a one-page summary of all the data from all the records. To do that, prevent the computer from adding blank lines at the end of each record. To prevent the blank lines, move the cursor to "Lines per page: 66", and change the 66 to 0.

Normally, the computer prints onto paper the title of each field and also its data, like this:

```
Last name: Reagan
First name: Ronnie
```

You can make the printer omit the titles and print just:

```
Reagan
Ronnie
```

To do that, move the cursor to "Print item names (Y/N): Y", and change the last Y to N. Make sure you change it to N, whenever you print mailing labels.

To print on the screen instead of on the printer, move the cursor to "Print to: PRN:", and change the "PRN:" to "CON:".

Mailing labels. Here's a review of the tricks you must use to print mailing labels in ZIP-code order.

For the print options, change the "Print item names" to N, and change the "Lines per page" to 6.

Use this form:

```
First name: +                          Last name: x
Address: x
City: +                                State: +          ZIP code: xs
```

LAYOUT STYLE

When you design a form, you can invent any field names you wish, and put them wherever you like on the screen. But if you follow these guidelines, PFS File will work better for you.

Specific fields. Make each field as specific as possible.

For example, do *not* have a field called "Person's name". Instead, have a field called "First name", a field called "Last name", and a field called "Middle initial", so that you can alphabetize on the basis of last name but also print messages greeting people on the basis of first name ("Dear Sam").

If you hire programmers, and you're keeping track of each programmer's abilities, do *not* have a field called "Which microcomputers the programmer knows". Instead, have a field called "Knows IBM PC?", a field called "Knows Macintosh?", a field called "Knows Apple 2e?", a field called "Knows Commodore 64?", etc. That way, if you're looking for a programmer to convert programs from the IBM PC to the Macintosh, you can search for programmers that have "Knows IBM PC?: Y" and "Knows Macintosh?: Y".

Index field. The field that begins the record is called the *index field*. It's the field in the upper-left corner of the record's main screen.

For example, if each record begins with a field called "Last name", then "Last name" is the index field.

The index field is more important than the other fields. Here's why. . . .

When you tell the computer to search for records that match fancy criteria (such as "age >21 and class <10"), the computer finds those records by scanning through the entire file, from beginning to end. If the file is long, the searching takes a long time.

But the computer can hop to the correct record *immediately*, if your search involves just the index field, and if the search doesn't involve "<" or ">" or ".." or "?".

For example, if the index field says "Last name", and you tell the computer to search for "Last name: Smith", the computer will hop to the Smiths immediately.

That's because, while you're adding records to the file, the computer automatically creates an index that tells where each record is located; and the index is organized on the basis of the record's index field.

To speed up most of your searches, figure out which field you'll search on most often, and make that field the index field. For example, if you're a doctor who usually searches through patient records on the basis of the patient's last name, make the index field be "Last name", by putting "Last name:" in the upper-left corner of the screen. That way, you'll be able to find your patients faster.

Wide fields. After you type the name of the field and its colon, leave lots of space before typing the name of the next field.

For example, if you have a field named "Age", do *not* leave just one or two spaces after it. If you leave just two spaces, you can handle one-digit and two-digit ages, but you can't handle ages such as 102 (which requires three spaces), and you can't search for everybody whose age is "not between 35 and 102", which is written "/=35..102", which requires nine spaces.

So leave lots of space for each field. Most of your screen should consist of blank spaces, so that you can occasionally write fancy searches (such as "/=35..102") and handle weird folks who have long names, such as my pet gorilla, "Alexandopoulosinskianonio Federiconacreon Szerchabiofskowitz, Jr." (His nickname is "Harry".)

Even though most of your screen should consist of blank spaces, PFS does *not* put all those blanks onto the disk. Instead, PFS puts on the disk just the data itself. So leaving lots of blanks on your screen does *not* consume space on the disk. PFS's ability to automatically remove the blanks from the data, before copying the data onto the disk, is called *data compression*.

Order of fields. Put the fields in the same order that you want them printed on paper.

For example, on mailing labels you want the first name to appear before the last name; so put the "First name" field before the "Last name" field.

What happens if you mainly look up people by last name (so that "Last name" should be your index field), but you also want to occasionally print mailing labels (for which "First name" should be your first field)?

Solution: make the index field be "Last name" (so that you can search for people quickly), but also create another field underneath, called "Full name for mailing". For example, the index field would say "Last name: Smith", and the field underneath it would say "Full name for mailing: Sue Smith".

Comment field. The bottom field should be called "Comments", so that you can add extra comments about anybody who's unusual. That field should be at the *bottom*, so that you can add as many comments as you please (by pressing the PAGE DOWN key repeatedly).

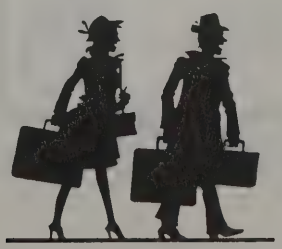
EXIT

When you've finished using PFS File, make the computer display the main menu. From that menu, choose "E. Exit". To do that, tap the E key, then the ENTER key.

The computer will exit from PFS File. The computer will say:

```
A>
```

Remove the disk from drive A. Then you can run a different program, or turn off the computer.



BEYOND PFS FILE

PFS REPORT

After buying PFS File, you'll want to buy PFS Report, which analyzes each file created by PFS File.

PFS Report prints a table that summarizes all the information in the file. Each record in the file becomes a row in the table; each field becomes a column.

The table looks like a spreadsheet. It's pretty.

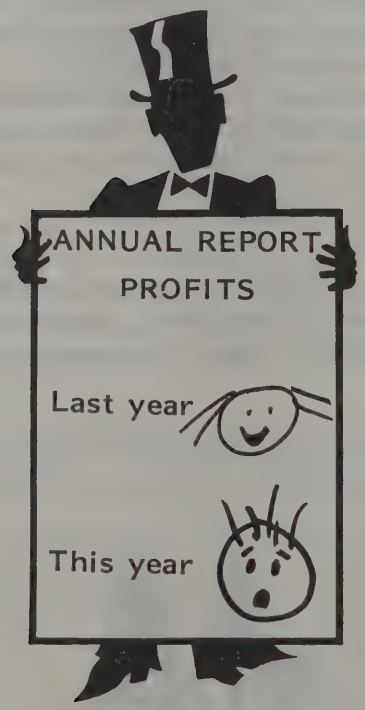
At the top of each column, PFS Report automatically prints an underlined heading. If the column contains numbers, PFS Report automatically lines up all their decimal points, so that the column looks straight, and automatically makes the column wide enough to hold the widest number.

If you don't want to see the whole report, you can request an abridgment, showing just the rows that interest you, and just the columns that interest you. You can also tell PFS Report to rearrange the columns, in whatever order pleases you.

If the leftmost column contains names, PFS Report automatically alphabetizes them, and rearranges the rows so that the names appear in alphabetical order. If several names in the leftmost column are the same as each other, PFS Report prints the name just once, so that all the rows about that name appear indented underneath the name.

If you wish, PFS Report will print totals at the bottom of each column, and also print averages. It can also print subtotals for groups of rows. It can also print extra columns, that show the results of performing mathematical calculations on the other columns, by using whatever formulas you wish.

The best version of PFS Report (version B for the IBM PC) lists for $125. Discount dealers charge just $67.



PC-FILE

Instead of buying PFS File & Report, you might prefer to buy *PC-File*. It runs on the IBM PC, the PC Junior, and clones.

Advantages of PC-File. The main reason for choosing PC-File is that it costs less. It was written by Jim Button in BASIC. You can get it from him for just $60, plus $5 for shipping. He calls his company "Buttonware Inc." (Box 5786, Bellevue, WA 98006, 206-454-0479, out-of-state 800-528-8866).

If you don't want to spend $65, you can get the program free, by copying it from computerized bulletin boards or computer user groups. Jim doesn't mind if you copy the program for your friends, provided you keep the copyright notice intact. (Jim gets his money from people who pay him for updates.) That method of marketing software is called *freeware* or *shareware*.

Besides being cheap, another reason for buying PC-File is that it's easy to use. It's even easier than using PFS File & Report.

The latest version of PC-File is called *PC-File 3, version 4*. In some ways, it's more powerful than PFS File & Report.

For example, PC-File lets you reverse the order of a pair of fields, so that you can look up "Adams, Jane" but print "Jane Adams" on her mailing label. (PFS File does *not* let you reverse the order of a pair of fields. PFS Report lets you reverse the order, but only in tables: that program doesn't deal with mailing labels.)

You can make PC-File find records more quickly than PFS, by telling PC-File to index *several* fields. (PFS builds an index only on the first field.)

PC-File can search phonetically. For example, if you tell PC-File to find all customers whose last name "sounds like Lewis", it will find all customers whose last name is Lewis or Louis.

Disadvantages of PC-File. Although PC-File has all those advantages over PFS File, it also has some drawbacks.

PC-File doesn't permit variable field widths. For example, if you say a certain field contains 40 characters, it will contain 40 characters in each record, and you can never vary that width. If you try to input more than 40 characters, the computer will refuse to accept the extra characters. If you input fewer than 40 characters, the computer will turn your input into 40 characters, by adding blank spaces afterwards, which waste space on your disk. (PFS File is more flexible: it lets you add attachments to your bottom field, and it does *not* put blank spaces onto your disk. Since it doesn't put blank spaces onto your disk, it wastes less space on the disk, and so it can squeeze more records onto a floppy disk than PC-File can.)

When you invent names for your fields, PC-File requires each name to be short: no more than 12 characters. So you can't have a field whose name is "Amount past due" (which contains 15 characters, including the spaces between the words); you must shorten it to "Amt past due". You must shorten "Number of surgical operations since 1980" to "# sur op 80s". (PFS File let the field name be as long as you like: up to 1680 characters, instead of 12.)

PC-File requires each record to be short: the entire record must fit on the screen. (PFS File permits the record to contain 32 screenfuls instead of just 1.)

PC-File restricts you to just 21 lines of the screen, and won't let you put more than 2 fields on a line. So you're limited to 42 fields. (PFS File permits 3200 fields per record instead of 42.)

Should you get PC-File? PC-File is quick, easy, and cheap, but restricts you to short records. It's adequate for handling mailing lists, but not for handling long personalized comments.

REFLEX

Of all the popular data-management systems, *Reflex* is the most fun. It's also one of the newest: it was invented in 1985. It runs on the IBM PC and clones.

It's lightning fast: just press a button, and the data you desire will flash on the screen immediately! You do *not* have to wait for your disk drive to spin!

When you're using Reflex, you can choose five ways to view your data. If you ask for a *form view*, Reflex will show you each record as a separate form, like PFS File. If you ask for a *list view*, Reflex will show you the entire file as a gigantic table, like 1-2-3. If you ask for a *graph view*, Reflex will graph the data, offering you the same five options as 1-2-3: pie chart, bar, stacked bar, line, or X-Y scatter plot. If you ask for a *crosstab view*, Reflex will show you a table of totals, in the style used by statisticians. If you ask for a *report view*, Reflex will print the results on paper, as a report with pretty headings and totals and subtotals and averages (like PFS Report) or on mailing labels; before printing on paper, it will also show you a preview on the screen.

Moreover, Reflex can divide your screen into windows and show a different view in each window. For example, you can see forms, tables, and graphs on the screen simultaneously. And the windows interact with each other: if you change the file by retyping the data in one of the windows, the other windows instantly change also, to reflect the changes you made. For example, if one window contains numbers, and another window shows a graph, and you start retyping the numbers, the graph simultaneously changes in front of your eyes.

Reflex makes the screen of your IBM PC look like a Macintosh. Like the Macintosh, Reflex divides the screen into windows and lets you use a mouse, to bounce around the screen quickly. To choose a command from Reflex's menu, just point to the command, by moving the mouse to it. The mouse is especially handy when you want to move to a particular cell in a table, or when you want to widen one of the table's columns by using the mouse to "stretch" the column.

To achieve its lightning speed, Reflex keeps the entire file in the RAM, so that it doesn't have to wait for a disk to turn. When you start using Reflex, you must wait for it to copy the file from the disk to the RAM; but once it's in the RAM, Reflex never goes back to the disk again.

Reflex is a large program. The program itself consumes about 320K of RAM; in addition, you must buy enough RAM to hold the file. If your file's very short, you can get by with 384K of RAM altogether. If your file's longer, you need more RAM.

If your file's *very* long, Reflex can't handle it well, since Reflex uses PC-DOS, which can't handle more than 640K of RAM. Instead, you must tell Reflex to find the file on the disk and then grab a part that will fit in 640K.

Like PFS File, Reflex permits variable field widths: if you put short data into a wide field, Reflex will store only the data you typed; it will *not* put blank spaces afterwards, which would waste space. So if you buy 640K of RAM, you'll discover that Reflex can store a surprisingly large number of records in it.

Reflex used to be published by *Analytica*, which charged $495. But then Analytica was merged into a larger company, *Borland*, which is famous for amazingly low prices. Borland lowered Reflex's list price to $99. Discount dealers sell it for $59. But to make full use of Reflex, you'll also want to buy a mouse and lots of RAM.

Q&A

After you use PFS File and PFS Report for a while, you'll discover many little things about those programs that annoy you—nuisances—and you'll want to perform many kinds of data-manipulation that PFS File and PFS Report can't handle. Although you can "get by" with PFS File and PFS Report, you'll long for more.

A group of critics analyzed PFS File and PFS Report, made a list of everything they didn't like about those programs, and invented an improvement, called *Q&A*. When you first start using Q&A, it seems to be almost the same as PFS File; but as you play with Q&A more and more, you'll notice *hundreds* of little ways in which Q&A is better! The more you use Q&A, the more you'll love it, and you'll probably never want to return to PFS File again.

Q&A is wonderful. I recommend it very highly. It's the closest I've ever seen to a *perfect* program. If you plan to become a top-notch programmer someday, you should buy Q&A and examine it very carefully, because it's the ultimate example of a powerful program that's easy to use.

Q&A replaces PFS File, PFS Report, and the PFS Write word processor. It also incorporates all the pleasant block-handling commands of IBM Writing Assistant. It does nearly everything that those four programs do, and does a lot more also.

As a bonus, Q&A also includes a feature called the "Intelligent Assistant", which lets you type any question you wish, in ordinary English: the computer knows enough English grammar and vocabulary to understand the gist of your question, and then tries to answer it by analyzing all the data in your file! The Intelligent Assistant works somewhat slowly, as it tries to interpret your English; and if you use strange words, it might ask you for synonyms; but it saves you the trouble of learning computerese.

Q&A lists for $299. Discount dealers charge $219—which is about the same as their total charge for PFS File & PFS Report & PFS Write.

Q&A incorporates most of the good features of PC-File. Like PC-File, Q&A lets you reverse the order of fields on mailing labels, and lets you index multiple fields (though Q&A can*not* search phonetically).

Like Reflex, Q&A lets you easily switch between a form view (which looks like PFS) and a list view (which looks like a spreadsheet). Unfortunately, Q&A can*not* display both of those views on the screen simultaneously; and Q&A's won't let you update the data unless you're in a form view. Q&A can*not* produce a graph view or crosstab view; so if you need graphs or crosstabs, you must get Reflex. Like Reflex, Q&A is very fast—it's much faster than PFS File and PC-File. Though Reflex is slightly faster than Q&A, that's only because Reflex cheats by storing the whole file in RAM. Q&A does *not* cheat, keeps the file on the disk, and lets your file grow to be as large as the disk.

Q&A has one important feature that PFS File, PFS Report, PC-File, and Reflex all lack: Q&A can check whether your input is reasonable, and warn you if you make a mistake. Those other programs cannot: they won't gripe if you input that a person's age is 172, or that a person's weight is 985 pounds, or that a person sent a check for $.007, or that a person's sex is N (instead of M), or that a person's middle initial is 7 (instead of a letter). More important, they won't gripe if you accidentally say to print a paycheck for $1,000,000. When you start developing your file, you can teach Q&A what kinds of input errors to gripe about. So Q&A will protect your data-entry clerk from being unreliable.

But after warning the data-entry clerk, Q&A permits the clerk to ignore the warning, so that the clerk can create an exception to the general restriction.

Disadvantages of Q&A. Although Q&A is better than the PFS series in almost every way, you *might* prefer PFS, for the following reasons.

PFS runs on a wide variety of machines by IBM, Apple, Commodore, and Radio Shack. To use Q&A, you must buy an IBM PC or clone.

PFS File needs just 128K of RAM. Q&A requires 512K. (But Symantec, which makes Q&A, will sell you a card filled with 256K of extra RAM chips for just $50.)

PFS File alone—without PFS Report and PFS Write—costs just $75 from discount dealers, and might be enough to meet your needs, if your needs are minimal.

Q&A is slightly harder to learn how to use than PFS File (though the bit of extra effort is well worth the extra power that Q&A gives you).

PFS File allows the bottom field to be infinitely long, so that it can even be a multi-page essay. Q&A doesn't allow you that luxury: in Q&A, each field must be reasonably short.

POWERBASE

PFS File can't handle relationships between different files. For example, suppose two different files contain information about Joan Adams, and Joan Adams is an old-fashioned woman who gets married and changes her name. PFS File can't automatically change her name in both files; you must tell PFS File about her new name *twice*. A program that *can* handle relationships between different files is called a *data-base management system (DBMS)*; PFS File is not a DBMS. PFS File is a *data-management system* but not a data-*base* management system. Since PFS File can't handle more than one file at a time, it's called just a *simple filing system*.

PFS File & Report, PC-File, Reflex, and Q&A are all just simple filing systems. If you want something fancier—if you want a DBMS—consider *Powerbase*. It's the easiest DBMS to learn how to use.

Powerbase runs on the IBM PC and lists for $349. Discount dealers charge just $185.

Like Q&A, Powerbase will warn you if your input is unreasonable. But whereas Q&A will let you ignore a warning, Powerbase will *not*; Powerbase will *insist* that you retype the input, and will keep waiting for you to do so; it will *not* permit exceptions. (In that respect, Powerbase is authoritarian, whereas Q&A is permissive. Powerbase is more appropriate for data-bases about money and crimes; Q&A is more appropriate for data-bases about personal characteristics.)

Powerbase has most of the other features of Q&A, except that Powerbase lacks a word processor and an "Intelligent Assistant". But Powerbase has two disadvantages: the computer handles Powerbase very slowly (Q&A is very fast); and although Powerbase is the easiest DBMS, it's not as easy as Q&A. So get Powerbase only if you need a DBMS (instead of a simple filing system) or if you need authoritarian input-checking.

CORNERSTONE

Like Powerbase, *Cornerstone* is a DBMS that checks the reasonableness of your input and runs on the IBM PC.

Cornerstone is slightly fancier than Powerbase: Cornerstone lets you combine information from several files more easily. Another advantage of Cornerstone is that it runs faster.

Unfortunately, Cornerstone's menus are much harder to understand than Powerbase's, so I don't recommend Cornerstone to beginners.

Cornerstone used to list for $495. Recently, the list price was lowered to $99.95, but most dealers assume you haven't heard about the reduction and so they still charge several hundred dollars.

DBASE

All those programs—PFS File & Report, PC-File, Reflex, Q&A, Powerbase, and Cornerstone—use menus. You can't change their menus.

For example, every time you start running PFS File, you must look at this menu:

```
PFS FILE MAIN MENU

1. Design file
2. Add forms
3. Copy
4. Search/update
5. Print
6. Remove
E. Exit
```

The menu is inflexible: you can't change it.

If you'd like to invent your *own* menu, so that it's customized to fit the peculiar needs of your own application, you must write your own data-management program. To do that, you must learn a programming language.

You've already learned the programming language called "BASIC". Although BASIC's the best language for creating games, tutorials, scientific analyses, and simple business calculations, it's *not* the best language for creating data-management programs: instead, you should use a language called *DBASE*, because DBASE makes the computer understand the meaning of the words TOTAL, AVERAGE, SORT, REPORT, BROWSE, INDEX, and FIND, so that you can use those words in your programs. By using DBASE, you can create your own customized DBMS.

DBASE is published by Ashton-Tate. The original version was named *DBASE 2* instead of DBASE 1, because programmers don't trust version 1 of anything! DBASE 2 ran on CP/M computers, then was rewritten for the IBM PC. After DBASE 2 came an improvement called *DBASE 3* (for the IBM PC only), which included extra commands, was easier to learn how to use, and also ran faster. The newest version, called *DBASE 3 PLUS*, includes even more commands, is even easier to learn how to use, and runs even faster. DBASE 3 PLUS runs only on the IBM PC, but a Macintosh variation will be available soon.

DBASE 3 PLUS lists for $695; discount dealers charge $395. DBASE 3 is no longer sold. DBASE 2 is still available, for folks who can't afford DBASE 3 PLUS. Discount dealers sell DBASE 2 for just $259.

PFS File and DBASE 3 PLUS are the most popular tools for data management. PFS File is popular because it's so easy to learn; DBASE 3 PLUS is popular because it gives professional programmers so much power.

DBASE COMPETITORS

Like DBASE, *Paradox* and *RBASE 5000* include programming languages for creating your own DBMS. But Paradox's programming language is too hard to learn, and RBASE 5000's programming language doesn't contain enough commands.

If you refuse to learn a programming language, you might find Paradox and RBASE 5000 attractive, since they include nice menus that let you avoid programming. But to *totally* control the way the DBMS acts, you must write a program; and for programming, DBASE is your best bet. That's why DBASE far outsells Paradox and RBASE 5000, even though Paradox and RBASE 5000 are heavily advertised.

DBASE, Paradox, and RBASE are all being continually improved, to incorporate each other's features. Maybe in the future, Paradox and RBASE will improve their programming languages enough to seriously compete with DBASE. But by then, DBASE itself will probably have improved.

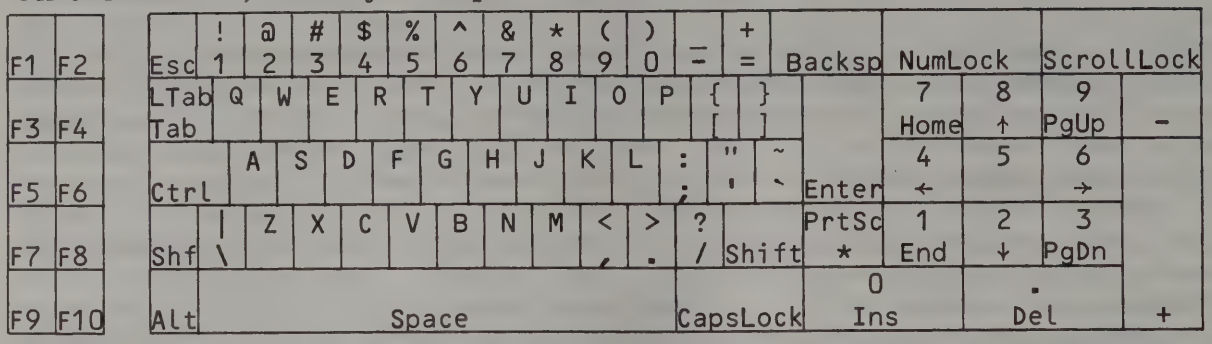
IF YOU DON'T HAVE AN IBM PC

If your computer isn't IBM-compatible and your needs are modest, get PFS File & Report. If your needs are *not* modest, and you find PFS File & Report inadequate, you should seriously consider switching to an IBM PC (or clone), so that you can use the best software. You'll also be switching to a faster disk system: PC-DOS lets the IBM PC read disks faster than most computers by Apple, Commodore, and Radio Shack.

WOW! YOU FINISHED CHAPTER 9: DATA-MANAGEMENT SYSTEMS!
TURN THE PAGE FOR APPENDIX 1: VERSIONS OF BASIC.

THE KEYBOARD

On the IBM PC, the keys are placed like this:


F1 F2 F3 F4 F5 F6 F7 F8 F9 F10
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Backsp
NumLock ScrollLock
LTab Tab Q W E R T Y U I O P { [}]
7 Home 8 ↑ 9 PgUp -
Ctrl A S D F G H J K L : ; " ' ~ ` Enter
4 ← 5 6 →
Shf | \ Z X C V B N M < , > . ? / Shift
PrtSc * 1 End 2 ↓ 3 PgDn
Alt Space CapsLock
0 Ins . Del +

In that diagram, I wrote the words "Shift", "Backsp", "Tab", and "Enter"; but if you look at your actual IBM PC keyboard, you'll see arrows on those keys instead. The SHIFT key shows an arrow pointing up; the BACKSPACE key shows an arrow pointing to the left; the TAB key shows arrows crashing into walls; and the ENTER key shows an arrow that's bent. IBM put arrows on those keys instead of words, to help people who don't read English.

On your IBM PC keyboard, find those keys, and double-check to make sure you found them correctly. . . .

The left SHIFT key is above the ALT key.
The right SHIFT key is above the CAPS LOCK key.
The BACKSPACE key is in the top row and left of the NUM LOCK key.
The TAB key is left of the Q.
The ENTER key is above the PRT SC key.

To type a number easily, use the keys in the top row, *not* the keys on the right side. (The keys on the right side produce numbers only if you tap the NUM LOCK key beforehand. If you've tapped the NUM LOCK key to make them produce numbers, and you want to *stop* making them produce numbers, just tap the NUM LOCK key again.)

GET STARTED

To start, you can choose either the "easy" way or the "fancy" way. The "easy" way is easier to understand; the "fancy" way lets your IBM do extra tricks.

If you choose the "easy" way, do *not* use a disk; remove any disks from the drives.

If you choose the "fancy" way, you need a disk drive, and you must make sure it contains a special kind of disk. The disk should be a "DOS system master" (which contains BASIC) or a copy of that disk. Put the disk into the *left* disk drive. When you put the disk into the drive, make sure the disk's label is on *top* of the disk, and make sure the disk's big oval cutout goes into the drive *before* the label does. Close the drive's door.

Attach the computer to a TV or monitor. Turn on the TV or monitor. (If you're using a TV, turn it to channel 33.) Then flip the computer's power switch to the ON position. (The power switch is red; it's on the right side of the computer.)

If you put your ear next to the computer, you'll hear its fan hum softly.

At first, the screen will display strange blips, as if it were broken. But after 8 seconds, the screen will become black, except for one blinking horizontal line in the upper left corner.

After about 30 seconds (the exact time depends on how much RAM your computer has), the computer will beep. Then the computer will print a message on the screen.

If you're using a TV (instead of a monitor), the message will be hard to read. Try channels 32, 33, 34, and 10; see which channel works best on your TV. Adjust your TV's fine-tuning knob. Also, adjust the contrast and brightness. Even after making all those adjustments, the writing will still look somewhat fuzzy; the computer will improve the writing if you give it a special command, which I'll explain later.

If you chose the "easy" way (without disks), skip to the next paragraph. If you chose the "fancy" way, the computer will say "Enter new date". (If your "DOS system master" disk is old, the wording might be slightly different.) Type the date. For example, if today is January 27, 1986, type 01-27-86 and then press the ENTER key. The computer will say "Enter new time". Type the time, using a 24-hour clock. For example, if the time is 1:45 AM, type 1:45 and then press the ENTER key; if the time is 1:45 PM, type 13:45 and then press the ENTER key; if the time is 7 seconds after 1:45 PM, type 13:45:07 and then press the ENTER key. The computer will say:

```
A>
```

Type the word "basica", like this:

```
A>basica
```

Then press the ENTER key.

Regardless of whether you chose the "easy" way or the "fancy" way, the computer will finally say "Ok".

If you're using a TV instead of a monitor, type:

```
width 40
```

That tells the computer to make the writing twice as wide, so that you can read it more easily.

HOW TO PROGRAM

If you're too lazy to type the word RUN, just tap the F2 key. When you tap the F2 key, the computer automatically types the word RUN for you, and automatically presses the ENTER key for you, and then automatically runs your program!

MAKE A MISTAKE?

If the computer gripes at you, press the ENTER key immediately.

After pressing the ENTER key, correct the error, by retyping the line that the computer griped about.

HOW THE COMPUTER REARRANGES YOUR PROGRAM

If you're too lazy to type the word LIST, just tap the F1 key. When you tap the F1 key, the computer automatically types the word LIST for you. After the computer types the word LIST, tap the ENTER key.

CLEARING THE SCREEN

Your keyboard lacks a CLEAR key. To clear the screen, hold down the Ctrl key; and while you keep holding down the Ctrl key, tap the HOME key.

Another way to clear the screen is to give the CLS command.

HOW TO PRINT ON PAPER

Your keyboard lacks a PRINT key. Instead of pressing a PRINT key, tap the PrtSc key *while holding down the SHIFT key*.

Your keyboard lacks an ECHO key. Instead of pressing an ECHO key, tap the PrtSc key *while holding down the Ctrl key*.

GO TO

The BREAK key is at the upper right-hand corner of the keyboard. It says SCROLL LOCK on it. To abort a program, hold down the CONTROL key, and while you keep holding down the CONTROL key, tap the BREAK key!!

DISKS

At the beginning of this IBM PC appendix, you'll find a section entitled "GET STARTED". It explains how to get started using the computer. It says there are two ways: an easy way, and a fancy way.

To use disks, you must start the computer by using the fancy way. For details, reread the section entitled "GET STARTED".

If you started the computer by using the easy way instead, turn the computer off and then restart the computer by using the fancy way.

How to turn a blank disk into an IBM disk. If you buy a blank disk and want the IBM computer to use it, you must turn the blank disk into an IBM disk.

To do that, put the blank disk into the disk drive that's on the right, then type:

```
system
diskcopy a: b:
```

The computer will say:

```
Insert source diskette into drive A:
Insert target diskette into drive B:
Strike any key when ready
```

Press any key on the keyboard. The computer will copy all information from the left drive's disk to the blank disk. The blank disk will become an IBM disk that's an exact copy of the other disk.

Then the computer will ask:

```
Copy another (Y/N)?
```

If you don't want to make another copy, type the letter n.

Finally, return to BASIC, by typing:

```
basica
```

How to choose a name. Your program's name can be short (such as JOE) or long: up to 8 characters. The name's first character must be a letter; the remaining characters should be letters or digits.

How to use drive B. The left drive is called *drive A*. The other drive is called *drive B*. Normally, the computer uses drive A. If you want JOE to be in drive B instead of in drive A, put a B and a colon before JOE, like this:

```
save "b:joe"
```

To make the computer print the names of all the programs in drive B, type this:

```
files "b"
```

Exception: if you're using an obsolete version of DOS (version 1 or version 1.05 or version 1.1), you must type this instead:

```
files "b:*.*"
```

Differences. Instead of saying RENAME "JOE" TO "FRED", say NAME "JOE" AS "FRED". Instead of saying KILL "JOE", say KILL "JOE.BAS", to emphasize that JOE was written in BASIC.

TAPES

Practically everybody who buys an IBM PC uses disks instead of tapes. Therefore, I won't bother explaining IBM's tape system, which practically nobody uses.

EDITING

The DELETE key says "Del" on it. The INSERT key says "Ins" on it.

PAUSING

Your keyboard lacks a PAUSE key. To make the computer pause, hold down the Ctrl key; and while you keep holding down the Ctrl key, tap the NUM LOCK key.

To stop pausing, press the SPACE bar.

To indent, you can hit the SPACE bar repeatedly. But to indent more easily, hit the TAB key (which is left of the Q key and has arrows on it).

ZONES

If you're using a monitor that prints 80 characters per line, your screen has five zones: zones 1-4 are each 14 characters wide; zone 5 is 24 characters wide.

If you said WIDTH 40 (because you're using a TV instead of a monitor), your screen has just two zones: zone 1 is 14 characters wide; zone 2 is 26 characters wide.

PIXELS

Before giving any commands about pixels, you should say:

```
screen 1
```

That arranges the screen so that the X coordinate can go from 0 to 319, and the Y coordinate can go from 0 to 199.

Whenever you finish using pixels, say:

```
screen 0
```

That makes the computer return to non-graphics.

Say PSET instead of PLOT.

To use the CIRCLE and PAINT commands, you must insert a DOS disk before turning on the computer. (Review the "GET STARTED" section of this IBM PC appendix.)

Colors. After saying SCREEN 1, you're limited to these colors:

0. black
1. cyan (greenish blue)
2. magenta (purplish red)
3. cream (yellowish white)

For example, if you type—

```
line (0,0)-(100,0),2
```

the computer will draw a line using color 2, which is magenta.

Changing the background. Normally, your drawing has a black background. But if you say—

```
color 1
```

everything on the screen that was black suddenly becomes blue. Moreover, any future drawing you try to do in black will be done in blue instead. That's because, in the COLOR command, 1 is the code number for blue. To choose your background, you can use any of the sixteen standard computer colors: 0=black, 1=blue, 2=green, 3=cyan, 4=red, 5=magenta, 6=brown, 7=cream, 8=gray, 9=light blue, 10=light green, 11=aqua, 12=pink, 13=light magenta, 14=yellow, 15=white.

Changing the palette. Normally, your drawing has a black background, and uses a *palette* consisting of three colors: color 1 is cyan, color 2 is magenta, and color 3 is white. That's called palette #1. If you wish, you can switch to palette #0, which consists of a different three colors: color 1 is green, color 2 is red, and color 3 is brown. To switch to palette #0, say:

```
color ,0
```

Remember to put the comma before the zero.

When you type that command, everything that was colored using palette #1 (cyan, magenta, white) suddenly changes to palette #0 (green, red, brown). In other words, everything that was cyan suddenly becomes green; everything that was magenta suddenly becomes red; and everything that was white suddenly becomes brown. If you do any more drawing afterwards, you'll be using palette #0: color 1 is now green, color 2 is now red, and color 3 is now brown.

To return to palette #1, say:

```
color ,1
```

You can choose the background color and the palette at the same time, like this:

```
color 1,0
```

That makes the background color be 1 (blue) and makes the palette be #0 (green, red, brown).

High resolution. To use pixels, you normally begin by saying SCREEN 1. If you say SCREEN 2 instead, you'll be using higher-resolution graphics, in which the pixels are thinner, and the X coordinate goes up to 639 instead of 319.

Unfortunately, if you say SCREEN 2, you must draw in just black and white: black is called "color 0"; white is called "color 1"; you can't use any other colors.

SOUNDS

To use the PLAY command, you must insert a DOS disk before turning on the computer. (Review the "GET STARTED" section of this IBM PC appendix.)

In the PLAY command, you can use the symbol ">" to mean "go up an octave", and you can use the symbol "<" to mean "go down an octave". For example, if you say—

```
play "g>cd<g"
```

the computer will play the note G, then go up an octave to play C and D in that higher octave, then go down to the original octave to play G again. Exception: if you're using an obsolete version of DOS (version 1 or 1.05 or 1.1), the symbols ">" and "<" don't work.

The lowest note the computer can play (which is the C in octave 0) is called *note 1*. The highest note the computer can play (which is the B in octave 6) is called *note 84*. To make the computer play note 84, type this:

```
play "n84"
```

To make the computer play its lowest note (1), then its middle note (42), then its highest note (84), type this:

```
play "n1n42n84"
```

RANDOM INTEGERS

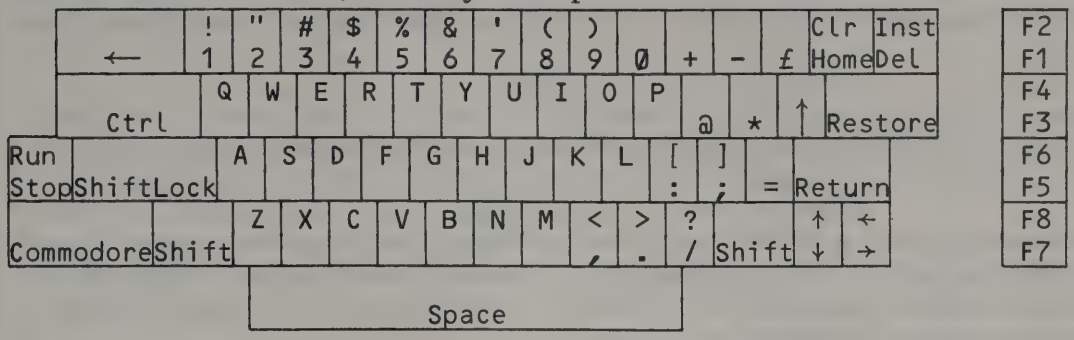
Instead of saying just RANDOMIZE, say RANDOMIZE TIMER. Exception: if your computer's using an obsolete version of DOS (version 1 or 1.05 or 1.1), you must say RANDOMIZE VAL(RIGHT$(TIME$,2)).

To get a random integer between 1 and 5, instead of saying RND(5), say 1+INT(RND*5).

COMMODORE 64

THE KEYBOARD

On the Commodore 64, the keys are placed like this:


← 1 ! 2 " 3 # 4 $ 5 % 6 & 7 ' 8 (9) 0 + - £ Clr Home Inst Del
Ctrl Q W E R T Y U I O P @ * ↑ Restore
Run Stop Shift Lock A S D F G H J K L : [;] = Return
Commodore Shift Z X C V B N M < , > . ? / Shift ↑ ↓ ← →
Space
F2 F1 F4 F3 F6 F5 F8 F7

The key in the bottom left corner is called the *Commodore key*. In the diagram, I wrote the word "Commodore" on that key; but if you look at your actual Commodore 64 keyboard, you'll see a fancy C on that key instead.

The zero key is Swedish.

The computer capitalizes all letters automatically; so to type a capital letter, do *not* press the SHIFT key.

There is no CAPS key.

The BACKSPACE key is marked DEL (which stands for DELETE). So to backspace, make sure you hit the DEL key. Do *not* hit the keys that show left-arrows; the left-arrow keys do *not* backspace correctly.

The ENTER key says RETURN on it.

GET STARTED

Attach the computer to a TV or monitor. The computer's power switch is on the computer's right side; flip that switch ON. Then turn on the TV or monitor.

If you're using a TV, turn it to channel 3 or 4. To switch the computer from channel 3 to channel 4 or back to channel 3, flip the computer's channel-selection switch, which is on the back of the computer. If the writing on the TV looks fuzzy, adjust the TV's fine tuning.

Adjust the screen's brightness and contrast.

When the computer's on and functioning correctly, the computer will say READY (instead of OK).

ARITHMETIC

To type a plus sign, do *not* press the SHIFT key. (If you accidentally press the SHIFT key, you'll be typing a crucifix instead of a plus sign.)

LONG COMMANDS ON SMALL SCREENS

A command can contain up to 80 characters. If you try to type more than 80 characters in a command, the computer will ignore the extra characters.

RANGES OF LINES

Your computer does *not* understand the word DELETE. To delete lines 30, 31, 32, 33, and 34, you can*not* type DELETE 30-34; instead, you must type:

```
30
31
32
33
34
```

COLONS

Although you can put several statements on the same line, the total number of characters in the line must not be greater than 80.

CLEARING THE SCREEN

The CLEAR key is marked "CLR HOME". To clear the screen, hold down the SHIFT key; and while you keep holding down the SHIFT key, press the "CLR HOME" key.

Your computer does *not* understand CLS. To make line 10 clear the screen, type this instead:

```
10 PRINT "♡";
```

To make that heart, hold down the SHIFT key with the CLR HOME key.

HOW TO PRINT ON PAPER

Your keyboard doesn't have a PRINT key or ECHO key, and your computer doesn't understand LPRINT or LLIST. To print on paper, here's what to do instead. . . .

Make sure the printer is plugged in, turned on, full of paper, and ready on-line.

To activate the printer, type this:

```
OPEN 4,4: CMD 4
```

That makes the computer use paper instead of your screen. Henceforth, the computer will print all its answers on paper instead of on your screen.

If you ever want the computer to print on your screen again, you must deactivate the printer, by typing:

```
PRINT#4: CLOSE 4
```

For example, to make the computer print just "I LOVE YOU" on paper, you must activate the printer, then tell the computer to print "I LOVE YOU", then deactivate the printer, like this:

```
OPEN 4,4: CMD 4
PRINT "I LOVE YOU"
PRINT#4: CLOSE 4
```

If a program prints on the screen but you want it to print on paper instead, activate the printer before you type RUN. Here's how:

```
OPEN 4,4: CMD 4
RUN
PRINT#4: CLOSE 4
```

Here's a fancy program that prints *two* poems. It prints "LOVE IS LIKE A DOVE" on your screen, and prints "LIFE IS FULL OF STRIFE" on paper:

```
NEW
10 PRINT "LOVE"
20 PRINT "IS LIKE A DOVE"
30 OPEN 4,4: CMD 4
40 PRINT "LIFE"
50 PRINT "IS FULL OF STRIFE"
60 PRINT#4: CLOSE 4
RUN
```

To LIST your program on paper instead of on your screen, type this:

```
OPEN 4,4: CMD 4
LIST
PRINT#4: CLOSE 4
```

GO TO

The BREAK key is marked "RUN STOP". So to abort a program, press the "RUN STOP" key.

If the computer is printing on the screen too quickly for you to read, you can make the computer print more slowly. To slow the printing, hold down the CTRL key while the computer prints. The computer will print slowly, as long as you keep holding down the CTRL key. When you lift your finger from that key, the computer will print quickly again.

LONG VARIABLE NAMES

Variable names must be short. The name of a numeric variable must be just 1 or 2 characters; the name of a string variable must be just 1 or 2 characters followed by a dollar sign. If you try to make the variable's name longer that that, the computer will handle it unreliably.

If you want a variable name that's 2 characters long, avoid the following 2-character names, which have special meanings: GO, TO, IF, ON, FN, TI, ST, OR.

STRING INPUT

Usually, you can abort a program by pressing the RUN STOP key. But while the computer is doing an INPUT statement, the RUN STOP key refuses to work, unless you simultaneously hold down the RESTORE key.

Test whether your Commodore is old. Commodore has sold two versions of the Commodore 64. The old version has trouble handling the INPUT statement; the new version handles the INPUT statement correctly. Commodore switched to the new version around the beginning of 1984. Test whether your Commodore 64 is new or old, by running this program:

```
10 INPUT "ARE YOU THE HAPPIEST PERSON IN THE WHOLE WORLD";A$
20 PRINT "YOU SAID ";A$
```

When you run the program, and the computer asks "ARE YOU THE HAPPIEST PERSON IN THE WHOLE WORLD", answer YES. If your Commodore 64 is new and works correctly, it will reply:

```
YOU SAID YES
```

If your Commodore 64 is old, it will say this nonsense instead:

```
YOU SAID ARE YOU THE HAPPIEST PERSON IN THE WHOLE WORLD? YES
```

What to do if your Commodore 64 is old. In an INPUT statement, the computer's question is called the *prompt*. An old Commodore 64 can't correctly handle a prompt longer than 38 characters. For example, this line's prompt's too long for an old Commodore 64:

```
10 INPUT "ARE YOU THE HAPPIEST PERSON IN THE WHOLE WORLD";A$
```

To prevent the old Commodore 64 from printing nonsense, shorten shorten the prompt to 38 characters or less, like this:

```
10 INPUT "ARE YOU THE HAPPIEST PERSON";A$
```

Better yet, eliminate the prompt altogether and replace it by a PRINT statement, like this:

```
10 PRINT "ARE YOU THE HAPPIEST PERSON IN THE WHOLE WORLD?"
11 INPUT A$
```

Many examples in this book contain prompts that are too long for an old Commodore 64 to handle. You must revise the examples or buy a newer computer.

STORIES

Reminder: some of the prompts are too long for an old Commodore 64 to handle, so you must revise them.

NUMERIC INPUT

If your Commodore 64 is old, shorten the prompts to 38 characters or less. If you forget to shorten a prompt, when you run the program and input a number the computer will say:

```
REDO FROM START
```

Abort the program (by holding down the RUN STOP and RESTORE keys simultaneously), then shorten the prompt.

FANCY CONDITIONS

If you try to invent your own examples using IF . . . THEN, make sure each line of your program contains no more than 80 characters. You'll be tempted to make some of the IF . . . THEN lines be longer, but longer lines won't work.

ELSE

Your computer doesn't understand the word ELSE.

DISKS

When you flip the disk drive's power switch, you create an electrical surge that damages any disk inside the drive. So before flipping the drive's power switch, make sure the drive is empty.

So to turn on your computer system, first remove any disk from the drive; then flip on the drive's switch; then turn on the computer. Memorize that sequence: "disks out, drive on, computer on."

When you're done using the computer, follow this sequence: "disks out, computer off, drive off".

How to insert a disk. To put a disk into the drive, first make sure the disk drive's red light is off. Then open the drive's door.

Insert the disk into the drive. When you insert it, make sure the disk's label is on *top* of the disk, and make sure the disk's big oval cutout goes into the drive *before* the label does. Close the drive's door.

How to turn a blank disk into a Commodore disk. If you buy a blank disk and want your computer to use it, you must turn the blank disk into a Commodore disk.

To do that, insert the blank disk into the drive (and close the door). To name the disk ELIZABETH and give it ID number 00, type this:

```
OPEN 15,8,15,"N:ELIZABETH,00"
CLOSE 15
```

Each disk you create must have a different ID number. The ID number can be any number from 00 to 99—or it can even be a pair of *letters*.

The drive's red light will glow for about 2 minutes. When the light stops glowing, your blank disk is finally a Commodore disk.

Differences. The Commodore 64 uses different disk commands than other computers:

Most computers	Commodore 64
`SAVE "JOE"`	`SAVE "JOE",8`
`LOAD "JOE"`	`LOAD "JOE",8`
`RUN "JOE"`	`LOAD "JOE",8` `RUN`
`FILES`	`LOAD "$",8` `LIST`
`DELETE "JOE"`	`OPEN 15,8,15,"S:JOE"` `CLOSE 15`
`RENAME "JO" TO "AL"`	`OPEN 15,8,15,"R:AL=JO"` `CLOSE 15`

How to choose a name. Your program's name can be short (such as JOE) or long: up to 16 characters.

Watch the red light. Whenever you give a disk command, the computer sends the command to the disk drive and immediately says READY. The word READY does *not* mean the drive's finished the command yet. The drive doesn't finish the command until the drive's red light stops glowing.

If you give a wrong disk command, the computer nevertheless sends it to the drive and says READY. But when the drive receives that command, the drive gripes by *blinking* its red light. The drive's light goes on, then off, then on, then off, then on, etc. So a blinking light means you made an error. A continually glowing light means your command is okay and the drive is obeying it. Here's the rule: "Glow is good. Blink is bad."

If you give a bad command and the light starts to blink, you must stop the blinking, by typing this:

```
OPEN 15,8,15,"I"
CLOSE 15
```

If you try to copy JOE onto a disk (by saying SAVE "JOE",8) and the disk *already* contains a program named JOE, the drive will gripe, by blinking its light. To avoid such griping, give this command instead:

```
SAVE "@:JOE",8
```

In that command, the at-sign followed by the colon tells the drive: save JOE, *even though the disk already contains a JOE.*

TAPES

To copy a program onto a tape, position the tape and notice the counter's number. Then type the word SAVE. At the end of that word, press the RETURN key.

Your computer will say:

```
PRESS RECORD & PLAY ON TAPE
```

Obey the computer: press the recorder's RECORD and PLAY buttons at the same time as each other.

The computer will say:

```
OK
SAVING
READY.
```

Then press the recorder's STOP button.

How to read a program from a tape. Put the tape into the recorder. (To do that, you might have to press the recorder's EJECT button.)

By using the REWIND, FAST FORWARD, PLAY, and STOP buttons (and the counter, if you have one), position the tape to the program's beginning (or slightly before). Then type the word LOAD. At the end of that word, press the RETURN key.

The computer will say:

```
PRESS PLAY ON TAPE
```

Obey the computer, by pressing the recorder's PLAY button.

The computer will say:

```
OK
SEARCHING
FOUND
LOADING
READY.
```

Then press the recorder's STOP button.

Type the word LIST or RUN; then press the RETURN key.

EDITING

To edit line 30, first put line 30 onto the screen, by saying LIST 30. (Say LIST, not EDIT: you computer doesn't understand the word EDIT.)

Next, move the cursor to the part of the line you want to correct, by pressing the pair of arrow keys at the lower-right corner of the keyboard. (By pressing those keys, with or without the SHIFT key, you can move the cursor in all four directions.)

When you've moved the cursor to the part of the line you want to correct, and make your corrections.

To delete a character, move the cursor just past that character, then press the DELETE key. The DELETE key deletes the character to the left of the cursor.

To replace a character, move the cursor to the character your want to replace, then type the new character you want instead.

To insert an extra character in the middle of the line, move the cursor to where you want that character to appear, then tap the INST key *while holding down the SHIFT key*, then type the character you want to insert. (To insert *two* characters, you must tap the INST key with SHIFT, then type the first character, then tap the INST key with SHIFT again, then type the second character.)

When you've finished correcting the line, press the RETURN key, which tells the computer to take the corrections seriously.

After pressing the RETURN key, you might like to move the cursor to the bottom of the screen. If you'd like to do that, press the down-arrow key several times.

Hassle. That editing procedure is easy, except for one hassle: under certain unusual conditions, the cursor keys and INST/DEL key don't work! Under those conditions, the cursor keys and INST/DEL keys print strange symbols on your screen instead of moving the cursor. If that happens to you, press the RETURN key and then try again to edit the line. If it happens again, press the RETURN key, then move the cursor to the bottom of the screen, and type the whole line over again.

I said the hassle occurs "under certain unusual conditions". Specifically, it occurs only if you begin an editing sequence (by pressing the cursor keys and INST/DEL key) when you're at the right-hand part of a line: the part that's to the right of an unmatched quotation mark or to the right of an inserted space (inserted by the INST key).

RENUMBERING

Your computer doesn't understand the word RENUM.

PAUSING

Your keyboard lacks a PAUSE key. Instead of pausing, you must either slow the computer down (by holding down the CTRL key) or abort (by pressing the STOP key).

After aborting a RUN, you can make the computer continue where it left off, by typing:

```
CONT
```

The word CONT will make the computer continue an aborted RUN, but will *not* make it continue an aborted LIST.

APOSTROPHE

Instead of an apostrophe, type a colon followed by the word REM (which stands for "remark"). For example:

```
40 C=40: REM BECAUSE RUSS HAS 40 COMPUTERS
```

LISTS OF DATA

If you invent your own list of data, remember that each line of your program must be short: no more than 80 characters!

COUNTING TO 100

If you indent the lines between FOR and NEXT, your computer ignores the indentation. When you LIST your program, you won't see any indentation. So don't bother to indent.

ZONES

Your screen has 4 zones. Each zone is 10 characters wide, so the entire screen is 40 characters wide.

TAB

On other computers, the leftmost character in the line is said to be at position 1. But on your computer, the leftmost character in the line is said to be at *position 0* instead; after position 0 come positions 1, 2, 3, etc.

So on your computer, if you say TAB(0), you'll be going to the leftmost position; if you say TAB(1), you will *not* be going to the leftmost position.

LOCATE

Your computer doesn't understand the word LOCATE. To make your computer move the cursor, use the following trick instead. . . .

In the middle of your program, write a PRINT statement that prints a string. (To do that, type a line number, followed by the word PRINT, followed by a quotation mark, followed by a string of characters, followed by another quotation mark.) But while you're in the middle of typing the string of characters, press a cursor-arrow key several times. When you press the cursor key, the cursor will *not* move immediately; instead, you'll see funny symbols in the middle of the string.

Later, when you RUN the program and the computer comes to that PRINT statement and sees those funny symbols, the computer will automatically move the cursor.

Here's the rule: while you're typing a string, if you try to move the cursor, the computer won't move the cursor until later when the program is running.

So to write a program that moves the cursor, tap the cursor-arrow keys in the middle of a string in a PRINT statement.

PIXELS

To make your computer easily handle pixels, you must insert a cartridge called *Simon's BASIC*, which costs extra. For details about how to use that cartridge, read the manual that comes with the cartridge.

SOUNDS

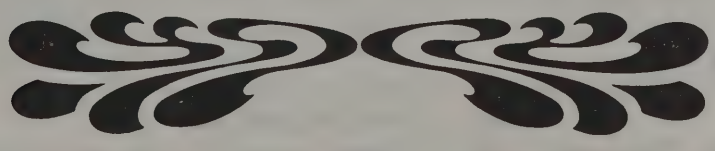
To make your computer easily produce sounds, you must buy the Simon's BASIC cartridge. For details about how to use that cartridge, read the manual that comes with the cartridge.

RANDOM INTEGERS

Instead of saying RANDOMIZE, say R=RND(0). To get a random integer from 1 to 5, instead of saying RND(5), say 1+INT(RND(1)*5).

RANDOM DECIMALS

For a random decimal between 0 and 1, say RND(1) instead of just RND.



ATARI 800 XL

CHOOSING A LANGUAGE

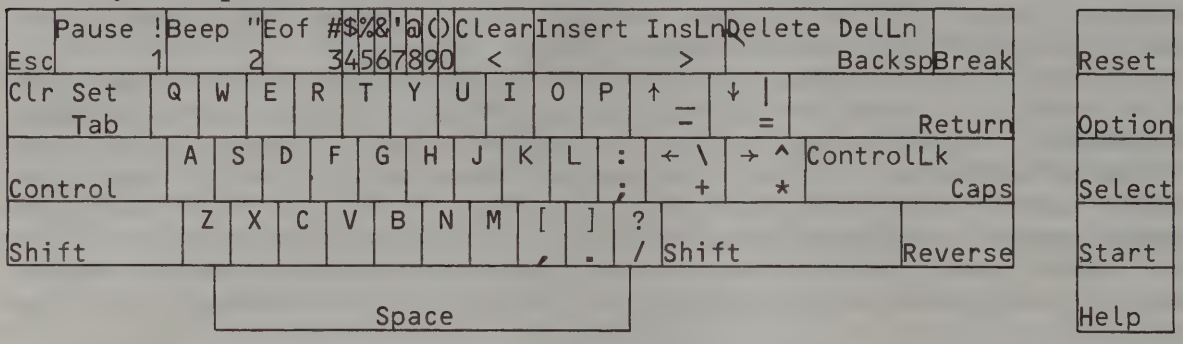
The Atari 800 XL understands *Atari BASIC*. Unfortunately, that version of BASIC gives you some hassles.

If you expect to do lots of programming, I recommend that you put into the computer a ROM cartridge called *Microsoft BASIC 2*, which provides a much better version of BASIC. Unfortunately, that cartridge costs about $55, and the average Atari owner is too cheap to buy it.

I'll explain *both* versions of BASIC: Atari BASIC and also Microsoft BASIC 2.

THE KEYBOARD

The keys are placed like this:



One of the keys looks like this:

If you tap that key, you'll be typing a plus sign. If you tap that key *while holding down the SHIFT key*, you'll be typing a backslash. If you tap that key *while holding down the CONTROL key*, you'll activate the left-arrow.

The computer capitalizes all letters automatically; so to type a capital letter, do *not* press the SHIFT key.

The ENTER key says RETURN on it.

GET STARTED

To begin, make sure the computer is turned OFF. (The power switch is at the computer's left rear.) If you have the Microsoft BASIC 2 cartridge, put it in the slot at the top of the computer. If you have a disk drive, turn it on, wait until the red BUSY light goes off, and then insert a disk: if you have Microsoft BASIC 2, insert the Microsoft BASIC 2 disk; otherwise, insert an Atari DOS disk; then close the drive's door. Turn on the computer.

Attach the computer to a TV or monitor. If you're using a TV, turn it to channel 2 or 3. To switch the computer from channel 2 to channel 3 or back to channel 2, flip the channel-selection switch, which is on the box attached to the back of the computer. If the writing on the TV looks fuzzy, adjust the TV's fine tuning. Adjust the screen's brightness and contrast.

If you're using Atari BASIC, the computer will say READY. If you're using using Microsoft BASIC 2, the computer will print the symbol ">". The "READY" or ">" means everything is OK.

HUGE & TINY NUMBERS

If you're not using Microsoft, the highest number the computer can handle is 9.99999999E97, and the tiniest decimal is 1E-98.

MAKE A MISTAKE?

If you're not using Microsoft, you can make these 21 major errors:

1. *You said to use some equipment. But that equipment's power isn't on.*
2. *You said to do something. But the computer doesn't have enough RAM.*
3. *You mentioned a number. But that number is too low—or too high.*
4. *You mentioned a variable. But your program contains too many variables.*
5. *You mentioned a string. But the string's longer than the DIM statement said.*
6. *You said READ. But the computer can't find enough DATA to read.*
7. *You mentioned a number. But that number ought to be between 1 and 32768.*
8. *You inputted a string. But the INPUT statement said you'd input a number.*
9. *You mentioned a subscript. But that subscript is too big or is inconsistent.*
10. *You gave a long list of GOSUBs or a long computation. But it's too long.*
11. *You said to compute. But the answer is a number that's too large or too small.*
12. *You said to go to a different line. But the computer can't find that line.*
13. *You said NEXT. But the computer can't find the FOR that matches the NEXT.*
14. *You gave a long instruction. But that instruction is too long.*
15. *You said NEXT or RETURN. But you erased the related FOR or GOSUB.*
16. *You said RETURN. But the computer can't find the corresponding GOSUB.*
17. *You gave an instruction. But the computer's memory contains an error.*
18. *You mentioned a string. But one of the characters in that string is wrong.*
19. *You said to LOAD a long program. But you didn't buy enough RAM to hold it.*
20. *You mentioned a device number. But that number should be between 0 and 7.*
21. *You said to LOAD. But the computer can't load that kind of file.*

For example, suppose line 50 says to do something, but the computer doesn't have enough RAM. The computer will say:

```
ERROR-  2 AT LINE 50
```

RANGES OF LINES

To say "LIST 30-80" or "LIST 30-" or LIST -80", you must use Microsoft. If you're not using Microsoft, say "LIST 30,80" instead of "LIST 30-80".

To use the DELETE command, you must use Microsoft and a disk drive, and you must say DEL instead of DELETE.

SPACES

If you're not using Microsoft, the computer will *not* automatically put spaces near numbers.

MULTIPLE CALCULATIONS

If you're not using Microsoft, replace each semicolon by this:

```
;" ";
```

For example, instead of saying—

```
PRINT 6+2;6-2;6*2;6/2
```

say this:

```
PRINT 6+2;" ";6-2;" ";6*2;" ";6/2
```

CLEARING THE SCREEN

The CLEAR key is next to the zero key.

Since the word CLEAR is written on the *top* part of the CLEAR key, you must simultaneously hold down the SHIFT key.

If you're not using Microsoft, say "GRAPHICS 0" instead of "CLS".

HOW TO PRINT ON PAPER

Your keyboard doesn't have a PRINT key or an ECHO key. Instead of saying LLIST, say:

```
LIST "P:"
```

If you use Microsoft, instead of saying LPRINT "I LOVE YOU" and LPRINT "VERY MUCH", type this:

```
Activate the printer.          OPEN#1, "P:" OUTPUT
Print onto paper, "I LOVE YOU". PRINT#1, "I LOVE YOU"
Print onto paper, "VERY MUCH".  PRINT#1, "VERY MUCH"
Deactivate the printer.        CLOSE#1
```

STRING VARIABLES

If you're not using Microsoft, the top line of your program must list all the string variables; for example, if your program uses the string variables G$, L$, and P$, the top line of your program should say:

```
1 DIM G$(40),L$(40),P$(40)
```

The number 40 lets each string contain up to 40 characters. If you know the string will contain fewer characters, you can use a smaller number instead; for example, if you know G$ will be the string "DOWN", which contains just 4 characters, you can say G$(4) instead of G$(40). Although 40 is large enough to handle most strings, if a string contains more than 40 characters you must type a larger number instead.

STRING INPUT

If you're not using Microsoft, you must put the question in a PRINT statement instead of an INPUT statement; instead of saying—

```
10 INPUT "WHAT IS YOUR NAME";N$
```

you must say—

```
10 PRINT "WHAT IS YOUR NAME";: INPUT N$
```

and also remember to say:

```
1 DIM N$(40)
```

ELSE

To say ELSE, you must use Microsoft.

DISKS

Your program's name must begin with a D and a colon, like this:

```
SAVE "D:JOE"
```

After the colon, you can type up to 8 letters and digits.

The computer will refuse to put your program onto the original Microsoft BASIC 2 disk; instead, put your program onto a *copy* of that disk.

Your computer doesn't understand the word FILES. To say KILL, you must use Microsoft. Say NAME instead of RENAME; and to use that concept, you must use Microsoft.

To turn a blank disk into an Atari disk, or make a copy of a disk, or make the computer print the names of all the programs on the disk, or do other fancy disk tricks, type the word DOS, and then follow the instruction manual that came with your disk operating system. (Atari has created several different versions of its disk operating system; each version has its own commands.)

TAPES

To copy a program onto a tape, adjust the tape recorder, then type the word CSAVE, then press the RETURN key twice. To retrieve that program, adjust the tape recorder, then type the word CLOAD, then press the RETURN key twice.

EDITING

Say LIST, not EDIT. The cursor doesn't blink. While using the DELETE, INSERT, and arrow keys to edit, you must hold down the CONTROL key.

To insert an extra character in the middle of the line, move the cursor to where you want that character to appear, then tap the INSERT key (while holding down the CONTROL key), then type the character you want to insert.

RENUMBERING

To say RENUM, you must use Microsoft and a disk drive.

PAUSING

To pause—or stop pausing—tap the 1 key while holding down the CONTROL key.

APOSTROPHE

If you're not using Microsoft, you must replace the apostrophe by a colon and REM, like this:

```
40 C=40: REM BECAUSE RUSS HAS 40 COMPUTERS
```

COUNTING TO 100

If you indent the lines between FOR and NEXT, your computer ignores the indentation. When you LIST your program, you won't see any indentation. So don't bother to indent.

To say just "NEXT" instead of "NEXT X", you must use Microsoft. If you're not using Microsoft, you must put a variable after the word NEXT.

FAVORITE COLOR

If you're not using Microsoft, and a FOR . . . NEXT loop contains a GO TO statement that makes the computer skip out of the loop, you should say "POP: GO TO" instead of "GO TO"; if the GO TO statement makes the computer skip out of *two* loops at once, say "POP: POP: GO TO" instead of "GO TO".

ZONES

The entire screen is 38 characters wide.

If you use Microsoft, the screen consists of 2 zones: the left zone is 14 characters wide; the right zone is 24 characters wide. If you're not using Microsoft, the screen consists of 4 zones: the each zone is 10 characters wide, except the rightmost zone, which is 8 characters wide.

TAB

To say TAB, you must use Microsoft.

LOCATE

Your computer doesn't understand the word LOCATE.

On your screen, the top line is called *line 0*. The leftmost position in a line is called *position 0*; positions 0 and 1 are so far to the left that they seem to be in the left margin.

Let's print the word "DROWN", beginning at the 7th position of line 3. If you use Microsoft, say:

```
PRINT AT (7,3) "DROWN"
```

If you're not using Microsoft, say:

```
POSITION 7,3: PRINT "DROWN"
```

PIXELS

Atari handles pixels differently from most other computers. Here are the details. . . .

Simple graphics. To use simple graphics, type:

```
GRAPHICS 3
```

That makes the computer split the screen into two parts. The top part of your screen will be used for graphics. The bottom of your screen will *not* be used for graphics: instead, it will be used for non-graphics (which Atari calls "text").

When you type GRAPHICS 3, the computer makes the top part of your screen (the graphics part) be very, very dark gray (i.e., black).

Your next step is to choose a color to draw in. You can choose these colors:

COLOR 0 *is black.*
COLOR 1 *is medium-light orange.*
COLOR 2 *is light green.*
COLOR 3 *is dark blue-turquoise.*

You must also choose which pixel you want to color: the pixel's X coordinate must be from 0 to 39; the pixel's Y coordinate must be from 0 to 19. For example, to make pixel (27,13) become light green, type this:

```
COLOR 2: PLOT 27,13
```

To make pixel (6,3) and pixel (25,19) both become medium-light orange, type this:

```
COLOR 1: PLOT 6,3: PLOT 25,19
```

If the colors on your TV look wrong, adjust your TV's COLOR and TINT dials.

Let's draw a dark blue-turquoise line from pixel (5,2) to pixel (20,17). If you use Microsoft, type this:

```
COLOR 3: PLOT 5,2 TO 20,17
```

If you're not using Microsoft, type this instead:

```
COLOR 3: PLOT 5,2: DRAWTO 20,17
```

The line looks jagged, because it's made of square pixels.

To print "WELCOME TO THE ART MUSEUM" on the bottom part of the screen (which is for non-graphics), just say:

```
PRINT "WELCOME TO THE ART MUSEUM"
```

Whenever you want to stop using graphics, type:

```
GRAPHICS 0
```

That tells the computer to use the entire screen for non-graphics (text).

You can put all those ideas into a program, like this:

```
10 GRAPHICS 3
20 COLOR 2: PLOT 27,13
30 COLOR 1: PLOT 6,3: PLOT 25,19
40 PRINT "WELCOME TO THE ART MUSEUM"
50 FOR I = 1 TO 4000: NEXT
60 GRAPHICS 0
```

Line 10 tells the computer to use simple graphics. Lines 20 and 30 draw a picture. Line 40 prints a caption. Line 50 makes the computer count to 4000: while the computer counts, you can admire the picture that the computer drew; if you want to admire the picture longer, change the 4000 to 10000. Line 60 returns the computer to its normal state, which is non-graphics.

In that program, lines 10, 50, and 60 are particularly important. Every simple graphics program should begin by saying GRAPHICS 3, and should end by saying "FOR I" and "GRAPHICS 0".

Smaller pixels. If you say "GRAPHICS 5" instead of "GRAPHICS 3", the pixels will be smaller, so that you can fit more pixels on the screen: the X coordinate can go from 0 to 79, and the Y coordinate can go from 0 to 39.

If you say "GRAPHICS 7" instead, the pixels will be even smaller: the X coordinate can go from 0 to 159, and the Y coordinate can go from 0 to 79.

If you say "GRAPHICS 15" instead, the pixels will be even smaller: the X and Y coordinates can each go from 0 to 159.

Setting different colors. The computer uses four colors. Normally, those colors are black, medium-light orange, light green, and dark blue-turquoise. But you can change them.

You can make the computer use any of these hues:

0. gray
1. gold
2. orange
3. red-orange
4. red
5. red-purple
6. purple
7. blue-purple
8. blue
9. blue-turquoise
10. turquoise
11. green-turquoise
12. green
13. green-russet
14. russet-green
15. russet

You can make the computer use any of these luminances:

0. very, very dark
2. very dark
4. dark
6. medium-dark
8. medium-light
10. light
12. very light
14. very, very light

To use those hues and luminances, you must learn about the registers:

Color number	Microsoft register	Non-Microsoft register
0	8	4
1	4	0
2	5	1
3	6	2

For example, suppose you want color 0 (which is normally black) to become light gold instead—and you're using Microsoft. Since "color 0" is Microsoft register 8, and since "gold" is hue 1, and since "light" is luminance 10, say:

```
SETCOLOR 8,1,10
```
register hue luminance

That makes the computer change color 0 to light gold. Everything on the screen that had been drawn in color 0 (which had been black) suddenly changes to light gold. And everything you draw in the future, by using color 0, will be light gold instead of black.

If you're not using Microsoft, color 0 is register 4 instead, so you must say instead:

```
SETCOLOR 4,1,10
```

In a typical Atari graphics program, line 10 says GRAPHICS 3 (or GRAPHICS 5 or GRAPHICS 7 or GRAPHICS 15); line 20 says SETCOLOR and changes a boring color (such as black) to a more interesting color (such as light gold); lines 30, 40, and 50 contain additional SETCOLOR statements.

SOUNDS

Instead of saying BEEP, say:

```
PRINT CHR$(253);
```

Your computer doesn't understand the word PLAY.

The SOUND command. Your computer handles the SOUND command differently from other computers. After the word SOUND, you should type four numbers, separated by commas, like this:

```
SOUND 0,127,10,4
```

That tells voice 0 to sing pitch 127 (which is "middle C"), using tone quality 10 (which is "pipe organ") and volume 4 (which is "medium-soft").

The computer will continue to play that sound, until you turn the sound off. To turn it off, press the RESET key (which is to the right of the BREAK key), or type the word END, or change the volume to 0, by typing:

```
SOUND 0,127,10,0
```

If you use Microsoft, you can also turn off the sound by pressing the BREAK key or typing the word STOP.

If you put a SOUND statement in the middle of a program, the computer will automatically turn off the SOUND when the program ends.

Controlling the duration. How can you make a sound last exactly 3 seconds? If you use Microsoft, just put 3*60 at the end of the SOUND command, like this:

```
SOUND 0,127,10,4,3*60
```

If you're not using Microsoft BASIC, say this instead:

```
SOUND 0,127,10,4: FOR I=1 TO 3*400: NEXT I: SOUND 0,127,10,0
```

Multiple voices. In this SOUND command—

```
SOUND 0,127,10,4
```

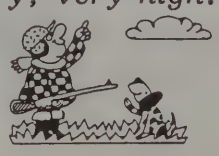
the 0 is the voice number; you can choose any number from 0 to 3. For example, if you type—

```
SOUND 0,127,10,4
SOUND 1,101,10,4
SOUND 2,84,10,4
SOUND 3,63,10,4
```

you'll hear voices 0, 1, 2, and 3 all sing simultaneously: you'll hear a chord.

Pitch. The pitch number (which is typically 127) can be any number from 0 to 255, but cannot be a decimal. Here are some popular pitches. . . .

255 is a deep bass note. It's "the C below middle C", "the C in the bass clef".
127 is "middle C".
63 is a soprano note. It's "the C above middle C", "the C in the treble clef".
31 is a very high note. It's "the C above the treble clef".
15 is very, very high. It's "the C above the C above the treble clef".
7 is very, very, very high, "the C above the C above the C above the treble".
3 is very, very, very, very high, "the C at the edge of your hearing range".
1 is very, very, very, very, very high, "a C that only your dog can hear."
0 is very, very, very, very, very, very high. Maybe not even your dog?



To translate traditional music notation into computer pitches, use this chart:

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
bass	255	241	227	214	202	191	180	170	160	151	143	135
alto	127	120	113	107	101	95	90	84	80	75	71	67
soprano	63	59	56	53	50	47	44	42	39	37	35	33
very high	31	29	28	26	24	23	22	20	19	18	17	16
very, very high	15	14	13		12	11	10		9		8	
very, very, very high	7		6			5			4			
edge of range of hearing	3					2						
dogs only	1											
not even dogs?	0											

(Note to musicians: to make full use of the Atari's range, I've tuned it slightly flat. For example, 127, which I'm calling "middle C", is slightly flatter than a piano's middle C; in fact, it's closer to the piano's B.)

Tone quality. In this SOUND command—

```
SOUND 0,127,10,4
```

the 10 is the tone quality ("pipe organ"). You can choose the following tone qualities: 0, 2, 4, 6, 8, 10, 12, or 14.

For most music, use 10. The 14 sounds almost the same as 10. The 12 is a buzz. The other tones are used for making the computer imitate guns, bombs, airplanes, trucks, motorcycles, and all the other evil joys of life.

Volume. In this SOUND command—

```
SOUND 0,127,10,4
```

the 4 is the volume. You can choose any volume from 0 to 15.

Volume 0 is silence. 1 is a barely audible whisper. 4 is soft and sweet and the best for most four-part harmony.

The higher the volume, the more unpleasant distortion you'll notice—especially if you try harmony.

PLAY. Your computer doesn't understand the word PLAY.

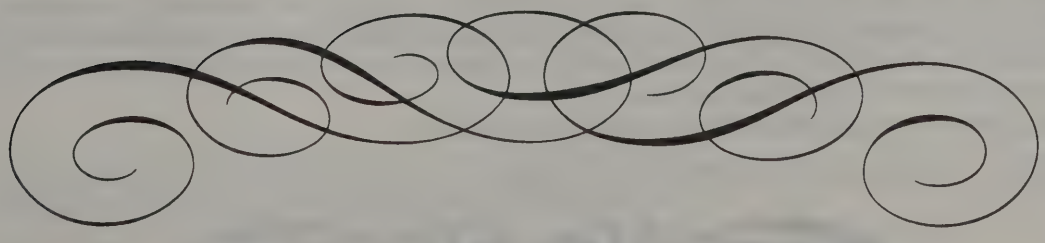
RANDOM INTEGERS

If you're not using Microsoft, you must make two changes:

Omit the word RANDOMIZE.
*Say 1+INT(RND(0)*5) instead of RND(5).*

RANDOM DECIMALS

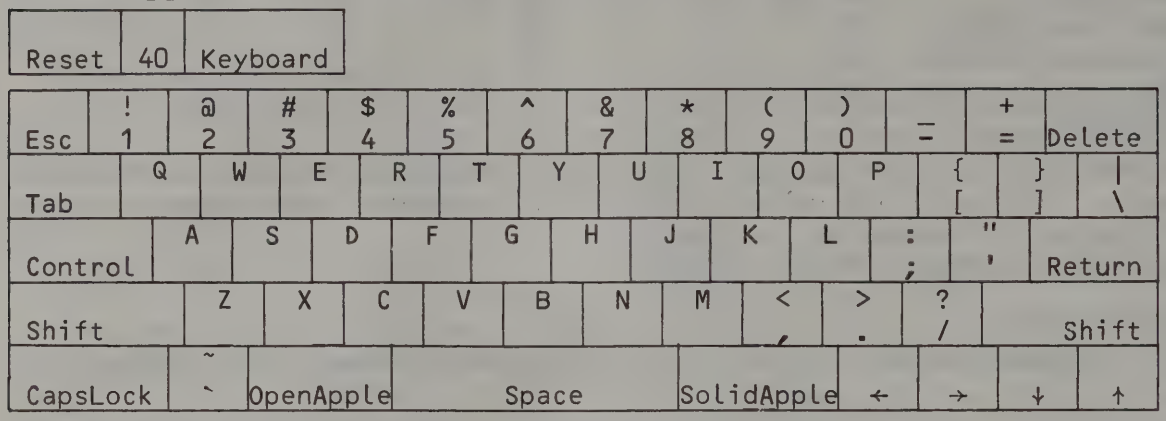
If you're not using Microsoft, say RND(0) instead of just RND.



APPLE 2C

THE KEYBOARD

On the Apple 2c, the keys are placed like this:



The keys next to the SPACE bar are called the *Apple keys*. In the diagram, I wrote the words "Open Apple" and "Solid Apple" on those keys; but if you look at your actual Apple 2c keyboard, you'll see pictures of apples on those keys instead.

There is no BACKSPACE key.

To correct an error, press the left-arrow key, until you get back to where the error was. Then correct the error, by retyping. Then, if you wish, press the right-arrow key to move to the right.

Your computer has a key marked DELETE, but it doesn't work when you're using BASIC.

The ENTER key says RETURN on it.

GET STARTED

Cut into the right side of your computer, you'll see a horizontal slit about 5 inches long. That slit is the entrance to the *built-in drive*.

At the center of the slit is a *door*, which is the same color as the keys on the keyboard. To open the door, push it to the left; then it will automatically pop up. To close the door, push it down. Practice opening and closing it.

Then open the door. Remove any disk from the built-in drive. Put the "System Utilities ProDOS disk" into the built-in drive; when inserting the disk, make sure the disk's label is on *top* of the disk, and make sure the disk's big oval cutout goes into the drive *before* the label does. After putting the disk into the drive, close the door.

Attach the computer to a TV or monitor.

The computer's green power switch is at the computer's rear, and has the numbers 0 and 1 on it. Flip the power switch on, by pressing the 1.

Turn on the TV or monitor.

If you're using a TV, turn it to channel 3 or 4. To switch the computer from channel 3 to channel 4 or back to channel 3, flip the channel-selection switch, which is on the box attached to the back of the computer. If the writing on the TV looks fuzzy, adjust the TV's fine tuning.

Adjust the screen's brightness and contrast.

The computer will put several messages onto the screen. Eventually, you'll see the *main menu*, which looks like this:

```
1.  Copy files
2.  Delete files
3.  Rename files
4.  Lock/unlock files
5.  Duplicate a disk
6.  Format a disk
7.  Identify and catalog a disk
8.  Advanced operations
9.  Exit system utilities
```

Tap the 9 key, then tap the RETURN key twice.

You'll see this symbol on the screen, at the upper-left corner:

```
]
```

That symbol's called a *bracket*. It means everything's OK. Your computer prints a bracket instead of the word OK.

You can make the characters on your screen be either thin or fat. The screen is wide enough to hold 80 thin characters per line, or 40 fat characters. Fat characters are easier to read, so choose fat characters if your eyesight is poor, or if you're using a cheap TV (which is blurry) instead of a monitor.

To choose fat characters (40 per line), tap the ESC key, then the 4 key. To choose thin characters (80 per line), tap the ESC key, then the 8 key.

RANGES OF LINES

Instead of saying DELETE, say DEL. To delete lines 30 through 80, type this:

```
del 30,80
```

SPACES

Your computer does *not* automatically put spaces near numbers.

For example, if you give this command—

```
print "the temperature is";4+25;"degrees"
```

the computer prints:

```
the temperature is29degrees
```

The computer does *not* automatically put a space before or after the 29.

To put spaces near the 29, put spaces in the strings nearby, like this:

```
print "the temperature is ";4+25;" degrees"
```

MULTIPLE CALCULATIONS

If you give this command—

```
print 6+2;6-2
```

your computer will print the answer to 6+2 (which is 8) and also print the answer to 6-2 (which is 4), but won't put any spaces between the answers. Your computer will print:

```
84
```

To force your computer to put a space between the answers, replace the semicolon by a *space in quotation marks*, like this:

```
print 6+2" "6-2
```

That makes your computer print:

```
8 4
```

CLEARING THE SCREEN

Your keyboard lacks a CLEAR key. Instead of pressing a CLEAR key, press the ESC key, then type an @ symbol. (To type the @ symbol, remember to press the SHIFT key.)

Your computer doesn't understand the CLS command. Instead of typing CLS, type the word HOME.

HOW TO PRINT ON PAPER

Your keyboard doesn't have a PRINT key or ECHO key, and your computer doesn't understand LPRINT or LLIST. To print on paper, here's what to do instead. . . .

Make sure the printer is plugged in, turned on, full of paper, and ready on-line. To activate the printer, type this:

```
pr#1
```

That makes the computer use paper instead of your screen. After you type PR#1, everything you and the computer type will appear on paper instead of on the screen.

If you ever want to use your screen again, you must deactivate the printer, by typing:

```
pr#0
```

For example, to make the computer print just "I love you" on paper, you must activate the printer, then tell the computer to print "I love you", then deactivate the printer, like this:

```
pr#1
print "I love you"
pr#0
```

If a program prints on the screen but you want it to print on paper instead, activate the printer before you type RUN. Here's how:

```
pr#1
run
pr#0
```

To LIST your program on paper instead of on your screen, type this:

```
pr#1
list
pr#0
```

In the middle of your program, one of the lines can say "pr#1". But if a line of your program simply says—

```
30 pr#1
```

the computer will get confused. Instead, say:

```
30 d$=chr$(4)
31 ?d$"pr#1"
```

To make line 60 deactivate the printer, say:

```
60 ?d$"pr#0"
```

For example, this program prints *two* poems. It prints "love is like a dove" on your screen, and prints "life is full of strife" on paper:

```
new
10 print "love"
20 print "is like a dove"
30 d$=chr$(4)
31 ?d$"pr#1"
40 print "life"
50 print "is full of strife"
60 ?d$"pr#0"
run
```

GO TO

To abort a program, hold down the CONTROL key; and while you keep holding down the CONTROL key, tap the C key.

If the computer is printing on the screen faster than you can read, you can make the computer slow down. To slow the printing, type this command:

```
speed=0
```

For example, to make the computer LIST your program slowly, and also PRINT slowly when it runs your program, type this:

```
speed=0
list
run
```

To make the computer print quickly again, at full speed, type this command:

```
speed=255
```

SPEED=0 means slow; SPEED=255 means fast. For an intermediate speed, pick a number between 0 and 255.

LONG VARIABLE NAMES

Variable names must be short. The name of a numeric variable must be just 1 or 2 characters; the name of a string variable must be just 1 or 2 characters followed by a dollar sign. If you try to make the variable's name longer than that, the computer will handle it unreliably.

If you want a variable name that's 2 characters long, avoid the following 2-character names, which have special meanings: TO, IF, ON, FN, GR, AT, OR.

STRING INPUT

In the INPUT statement, the computer's question is called the *prompt*. The prompt begins and ends with a quotation mark. On your computer, the prompt should end with *a question mark then a space then a quotation mark.* . . .

Most computers:
```
10 INPUT "WHAT IS YOUR NAME";N$
```
Apple 2c:
```
10 input "What is your name? ";n$
```

Usually, you can abort a program by tapping the C key while holding down the CONTROL key. But while the computer is doing an INPUT statement, that method won't work, unless you press the RETURN key afterwards.

FANCY CONDITIONS

Your computer can't handle the letter A followed by the word THEN. For example, it can't handle:

```
if x=a then print "wow"
```

To prevent the letter A from being next to THEN, reverse the equality—

```
if a=x then print "wow"
```

or use parentheses to separate the A from the THEN:

```
if (x=a) then print "wow"
```

ELSE

Your computer doesn't understand the word ELSE.

DISKS

If you buy a blank disk and want your computer to use it, you must turn the blank disk into an Apple disk.

To do that, turn the computer off, make sure the built-in drive contains the "System Utilities ProDOS disk", close the door, then turn the computer back on.

The screen will eventually show you the main menu, which looks like this:

```
1.  Copy files
2.  Delete files
3.  Rename files
4.  Lock/unlock files
5.  Duplicate a disk
6.  Format a disk
7.  Identify and catalog a disk
8.  Advanced operations
9.  Exit system utilities
```

Tap the 6 key, then tap the RETURN key four times.

The computer will say:

```
Place the disk you wish to format in the built-in drive.
Press RETURN to continue.
```

Obey the computer: put the blank disk into the built-in drive, then press the RETURN key.

The computer will grunt several times loudly: hold your ears! 20 seconds later, the computer will say:

```
Format complete.  Press RETURN to continue, ESC to return to the main menu.
```

The blank disk has been turned into an Apple disk.

Press the ESC key. You'll see the main menu again. Tap the 9 key, then tap the RETURN key twice. Then tap the ESC key, followed by the 4 key (for fat characters) or the 8 key (for thin characters).

You can*not* copy a program onto the System Utilities ProDOS disk. Instead, copy your program onto a blank disk that you turned into an Apple disk.

Differences. For the Apple 2c, omit quotation marks:

Most computers	Apple 2c
SAVE "JOE"	save joe
LOAD "JOE"	load joe
RUN "JOE"	run joe

Apple's notation differs in other ways also:

Most computers	Apple 2c
RENAME "JO" TO "AL"	rename jo,al
KILL "JOE"	delete joe
FILES	catalog

If you say—

```
cat
```

the computer will print an abridged catalog.

How to choose a name. Your program's name can be short (such as JOE) or long: up to 15 characters. The name's first character must be a letter; the remaining characters must be letters, digits, or periods.

How to use drive 2. The built-in drive is called *drive 1*. If you buy an extra drive, the extra is called the *external drive*; it's also called *drive 2*.

Normally, the computer uses drive 1. If you want it to use drive 2 instead, put a comma and a D2 at the end of the disk command. For example, to get a catalog of drive 2, say:

```
catalog,d2
```

To save JOE in drive 2, say:

```
save joe,d2
```

If you put a comma and D2 at the end of a command, the Apple assumes you're more interested in drive 2 than in drive 1. Therefore, it will automatically use drive 2 instead of drive 1 for all future commands. If you ever want to use drive 1 again, you must give a command that ends in a comma and D1.

How to copy an entire disk. To copy an entire disk, put the System Utilities ProDOS disk into the built-in drive. Turn the computer off, then on again. When you see the main menu, tap the 5 key, then tap the RETURN key three times.

The computer will say:

```
Place the source disk in the built-in drive.
Press RETURN to continue.
```

Obey the computer: into the built-in drive, put the disk you want to copy; then press the RETURN key.

The computer will say:

```
Place the destination disk into the built-in drive.
Press RETURN to continue.
```

Obey the computer: into the built-in drive, put a blank disk; then press the RETURN key twice.

The computer will turn the blank disk into an Apple disk. Then it will say, "Place the source disk into the built-in drive." Into the built-in drive, put the disk you want to copy; then press the RETURN key. Then it will say, "Place the destination disk into the built-in drive." Grab the blank disk that you had just turned into an Apple disk, and put it into the built-in drive; then press the RETURN key.

Seven more times, the computer will tell you to place the source disk and then the destination disk into the built-in drive. Obey the computer: swap the disks seven more pairs of times, and press the RETURN key after each time.

Finally, the computer will say:

```
Disk copy complete.  Press RETURN to continue, ESC to return to the main menu.
```

Press the ESC key. You'll see the main menu. Tap the 9 key, then the RETURN key twice, then say how wide you want the characters to be (by tapping ESC then 4, or ESC then 8).

How to create a quick-start disk. To create a quick-start disk (which helps start using BASIC faster), get a blank disk, and turn it into an Apple disk (by using the System Utilities ProDOS disk). Then get the main menu on the screen again, and make sure the System Utilities ProDOS disk is in the main drive. With the main menu still on the screen, tap the 1 key, tap the RETURN key four times, tap the down-arrow key six times, tap the right-arrow key, tap the down-arrow key again, tap the right-arrow key again, and tap the RETURN key.

The computer will say:

```
Place the destination disk in the built-in drive.
Press RETURN to continue.
```

Obey the computer: put the blank Apple disk into the built-in drive, then press the RETURN key. The computer will tell you to swap disks four more times. Finally, the computer will say:

```
Copying complete.  Press RETURN to continue, ESC to return to the main menu.
```

Press the ESC key. You'll see the main menu. Tap the 9 key, then the RETURN key twice, then say how wide you want the characters to be (by tapping ESC then 4, or ESC then 8).

You've created a quick-start disk. In the future, whenever you want to start using the computer quickly, put the quick-start disk into the drive (instead of the System Utilities ProDOS disk), and then turn on the computer. The computer will say:

```
PRODOS BASIC
]
```

Then just say how wide you want the characters to be (by tapping ESC then 4, or ESC then 8).

If you save a program on that disk, and call the program STARTUP (instead of JOE), the computer will automatically run that program whenever you start up the computer.

If you make the top two lines of that program say—

```
10 d$=chr$(4)
11 ?d$"pr#3"
```

the computer will automatically tap the ESC key and 8 key for you.

TAPES

The Apple 2c can't handle tapes.

EDITING

To edit line 30, first put line 30 onto the screen, by saying LIST 30. (Say LIST, not EDIT: your computer doesn't understand the word EDIT.)

On your computer, the cursor does *not* blink: it's a solid square that indicates where you'll be typing.

Your computer uses the Catholic method of editing, which uses the Catholic cross and its effect on bodies and souls. Here are the details. . . .

Tap the ESC key. A cross will appear in the middle of the cursor. Move the cursor (including its cross) up to the line you want to edit, by pressing the up-arrow key repeatedly. Then tap the ESC key again, to get rid of the cross. (The up-arrow key works only when you see a cross, because, according to Catholics, "to get to heaven you must follow the Cross". Most other keys are Temptations: they're effective only when you do *not* see a cross.)

Move the cursor to the part of the line you want to correct, by pressing the right-arrow key repeatedly.

Then make your corrections.

To replace a character, move the cursor to the character you want to replace, then type the new character you want instead.

To delete a group of characters, move the cursor to the first character in the group. Create a cross (by tapping the ESC key), then move the cross across the group of characters (by pressing the right-arrow key), until you've passed the last character in the group. Then get rid of the cross (by tapping the ESC key again). The characters that the cross passed over will go to heaven, and will disappear from their earthly existence. (Their bodies will remain on the screen, but their souls will have departed, so that the characters won't affect any future LIST.)

To insert extra characters, move the cursor to the place where you want the extra characters to begin appearing. Then move the cursor up (by tapping the ESC key, then the up-arrow key, then the ESC key). Type the characters you want to insert. Then create a cross (by tapping the ESC key), and move the cross to where you want the extra characters to begin appearing (by tapping the down-arrow and left-arrow keys). Finally, get rid of the cross.

When you've finished correcting the line, press the right-arrow key repeatedly, until you get *past* the last character in the line. Then press the RETURN key. To see what happened to the line, LIST it.

Because that Catholic method of editing is so long-winded, most programmers don't bother using it: instead, they type the entire line over!

RENUMBERING

Your computer doesn't understand RENUM.

PAUSING

Your keyboard lacks a PAUSE key. To make the computer pause, hold down the CONTROL key; and while you keep holding down the CONTROL key, tap the S key.

To stop pausing, press the SPACE bar.

APOSTROPHE

Instead of an apostrophe, type a colon followed by the word REM (which stands for "remark"). For example:

```
40 c=40: rem because Russ has 40 computers
```

COUNTING TO 100

If you indent the lines between FOR and NEXT, your computer ignores the indentation. When you LIST your program, you won't see any indentation. So don't bother to indent.

ZONES

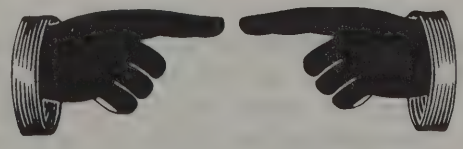
If you chose 80 characters per line (by tapping ESC and then 8), your screen has 5 zones, and each zone is 16 characters wide.

If you chose 40 characters per line (by tapping ESC and then 4), your screen has just 3 zones. The left zone is 16 characters wide; the middle zone is also 16 characters wide; the right zone is just 8 characters wide.

LOCATE

Your computer doesn't understand the word LOCATE. To print "DROWN" beginning at line 3's 7th position, say this:

```
vtab 3: htab 7: print "DROWN"
```



PIXELS

Apple handles pixels differently from most other computers. Here are the details. . . .

Simple graphics. To use simple, low-resolution graphics, type:

```
gr
```

That makes the computer split the screen into two parts. The top part of your screen will be used for graphics. The bottom of your screen will *not* be used for graphics: instead, it will be used for non-graphics (which Apple calls "text").

When you type GR, the computer makes the top part of your screen (the graphics part) be entirely black.

Your next step is to choose a color to draw in. You can choose these colors:

0. black
1. magenta (purplish red)
2. dark blue
3. lavender (light purple)
4. dark green
5. gray
6. blue
7. light blue
8. brown
9. orange
10. gray
11. pink
12. green
13. yellow
14. aqua (light greenish blue)
15. white

You must also choose which pixel you want to color: the pixel's X and Y coordinates must each be from 0 to 39. For example, to make pixel (31,37) become pink, type this:

```
color=11: plot 31,37
```

To make pixel (5,9) and pixel (17,21) both become yellow, type this:

```
color=13: plot 5,9: plot 17,21
```

If the colors on your TV look wrong, adjust your TV's COLOR and TINT dials.

To draw a green horizontal line from pixel (4,19) to pixel (8,19), type this:

```
color=12: hlin 4,8 at 19
```

To draw a blue vertical line from pixel (4,20) to pixel (4,30), type this:

```
color=6: vlin 20,30 at 4
```

To print "welcome to the art museum" on the bottom part of the screen (which is for non-graphics), just say:

```
print "welcome to the art museum"
```

Whenever you want to stop using graphics, type:

```
text: home
```

That tells the computer to use the entire screen for non-graphics (TEXT).

You can put all those ideas into a program, like this:

```
10 home: gr
20 color=11: plot 31,37
30 color=13: plot 5,9: plot 17,21
40 color=12: hlin 4,8 at 19
50 color=6: vlin 20,30 at 4
60 print "welcome to the art museum"
70 for i = 1 TO 4000: next
80 text: home
```

Line 10 includes the word HOME, to make sure that the bottom part of the screen (for non-graphics) begins entirely black. Lines 20-50 draw a picture. Line 60 prints a caption. Line 70 makes the computer count to 4000: while the computer counts, you can admire the picture that the computer drew; if you want to admire the picture longer, change the 4000 to 10000. Line 80 returns the computer to its normal state, which is non-graphics.

In that program, lines 10, 70, and 80 are particularly important. Every low-resolution graphics program should begin by saying "HOME: GR", and should end by saying "FOR I" and "TEXT: HOME".

High-resolution graphics. High-resolution graphics differs from low-resolution graphics in the following ways . . .

For low-res, you say "GR". For high-res, you say "HGR: VTAB 21".

For low-res, you say "COLOR=" and choose a color from 0 to 15. For high-res, you say "HCOLOR=" and choose one of these high-res colors:

0. *black*
1. *usually green; black if X is even; white if adjacent pixel isn't black*
2. *usually purple; black if X is odd; white if adjacent pixel isn't black*
3. *usually white; green if X is odd and adjacent pixel is black; purple if X is even and adjacent pixel is black*
4. *black*
5. *usually orange; black if X is even; white if adjacent pixel isn't black*
6. *usually blue; black if X is odd; white if adjacent pixel isn't black*
7. *usually white; orange if X is odd and adjacent pixel is black; blue if X is even and adjacent pixel is black*

For low-res, you say "PLOT". For high-res, you say "HPLOT".

For low-res, the X coordinate and Y coordinate each go from 0 to 39. For high-res, the X coordinate goes from 0 to 279, and the Y coordinate goes from 0 to 159.

For low-res, you say "HLIN" to draw a horizontal line, and you say "VLIN" to draw a vertical line. For high-res, you can draw a line in *any* direction, by saying "HPLOT TO". For example, if you say—

```
hplot 250,140
hplot to 70,90
hplot to 180,114
```

the computer will plot the pixel (250,140), then draw a diagonal line from there to pixel (70,90), then draw a diagonal line from pixel (70,90) to pixel (180,114). You can combine those commands into a single command:

```
hplot 250,140 to 70,90 to 180,114
```

Here's a complete program:

```
Use high-res.          10 home: hgr: vtab 21
Draw green lines.      20 hcolor=1: hplot 250,140 to 70,90 to 180,114
Write a caption.       30 print "welcome to the art museum"
Pause.                 40 for i = 1 to 4000: next
End the graphics.      50 text: home
```

SOUNDS

Instead of saying BEEP, say:

```
print chr$(7);
```

Your computer doesn't understand the words SOUND and PLAY.

RANDOM INTEGERS

Omit the word RANDOMIZE. To get a random integer from 1 to 5, instead of saying RND(5), say 1+INT(RND(1)*5).

RANDOM DECIMALS

For a random decimal between 0 and 1, say RND(1) instead of just RND.

APPLE MACINTOSH

CHOOSING A LANGUAGE

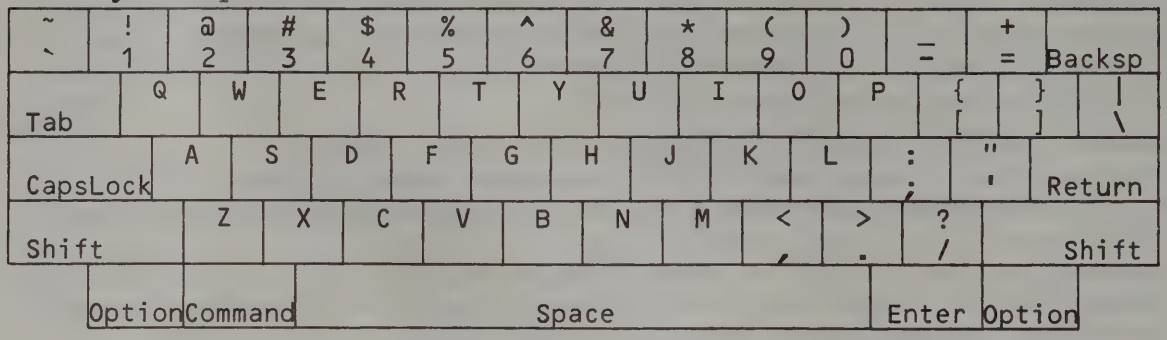
When you buy Apple's Macintosh computer, make sure you also buy *Microsoft BASIC*, which comes on a disk and costs extra.

Make sure you get *Microsoft BASIC version 2*; version 1 is obsolete. To tell which version you have, look at the label on the disk: if the label says "014096.200", you have version 2; if it says "014096.100", you have version 1.

The following comments apply to version 2.

THE KEYBOARD

The keys are placed like this:



To the left of the SPACE bar, you'll see the COMMAND key. In the diagram, I wrote the word "Command" on that key; but if you look at your actual Macintosh keyboard, you'll see a picture of a highway's cloverleaf on that key instead.

The keyboard contains a RETURN key and also an ENTER key. When you're using BASIC, those keys do the same thing as each other. You'll probably prefer the RETURN key, since your pinky can reach it more easily.

GET STARTED

Underneath the screen, you'll see a horizontal slit, 3½ inches long. Put the Microsoft BASIC disk into the slit. When you insert the disk, make sure the disk's label is on *top* of the disk, and make sure the disk's metal slider goes into the drive *before* the label does. Push the disk all the way in.

The computer's power switch is at the computer's rear, and has the numbers 0 and 1 on it. Flip the power switch on, by pressing the 1.

23 seconds later, you'll see an arrow in the upper left corner of the screen, and the words "Microsoft BASIC" near the upper right corner. Just above the words "Microsoft BASIC", you'll see a little picture of a disk. Little pictures are called *icons*. The picture of the disk, above the words "Microsoft BASIC", is called the *Microsoft BASIC disk icon*.

If you have difficulty reading the screen, adjust the brightness knob, which is at the front of the computer, near the bottom left, just under the overhang.

Using the mouse. Your computer comes with a *mouse*, which is about the size of a big rat, but squarish, so that it looks like a pack of cigarettes.

The mouse's *tail* is a wire that runs from the mouse to the computer. The area where the tail meets the mouse is called the mouse's *ass*.

The mouse's underside—its belly—has a hole it in, and a ball in the hole. That ball is called the *belly button*.

Put the mouse on your desktop and directly in front of you. Make the mouse lie flat (so that its belly button rubs against the desk). Make the mouse face you (so that you don't see its ass.)

Move the mouse across your desk. As you move the mouse, remember to keep it flat (so that its belly button rubs against the desk), and remember to keep the mouse facing you (so that you don't see its ass.)

As you move the mouse, the arrow on the screen moves also. If you move the mouse to the left, the arrow on the screen moves to the left. If you move the mouse to the right, the arrow on the screen moves to the right. If you move the mouse toward you, the arrow on the screen moves down. If you move the mouse away from you, the arrow on the screen moves up.

Practice moving the arrow, by moving the mouse. Remember to keep the mouse facing you at all times.

If you want to move the arrow far, and your desktop is small, move the mouse to the edge of the desktop, then lift the mouse off the desk, drop the mouse gently onto the middle of the desk, and then rub the mouse across the desktop in the same direction as before.

Getting into BASIC. Move the arrow to the middle of the Microsoft BASIC disk icon. To move the arrow, use the mouse. Make sure you move the arrow to the icon (the picture of the disk), *not* to the words "Microsoft BASIC" underneath it.

Moving the arrow to the middle of an icon is called *pointing at the icon*.

On the top of the mouse is a rectangular button that you can press. Pressing that button is called *clicking*. Pressing that button twice, in rapid succession, is called *double clicking*.

With the arrow still pointing at the icon, try double clicking: press the button twice in rapid succession. If you do it correctly, the icon will explode.

If the icon doesn't explode, it's because the arrow wasn't in the middle of the icon, or you paused too long between the two clicks, or you paused too briefly: try again!

After you've exploded the icon, you'll see a *window* on the screen. At the top of the window is a message saying "Microsoft BASIC, 5 items, 278K in disk, 21K available".

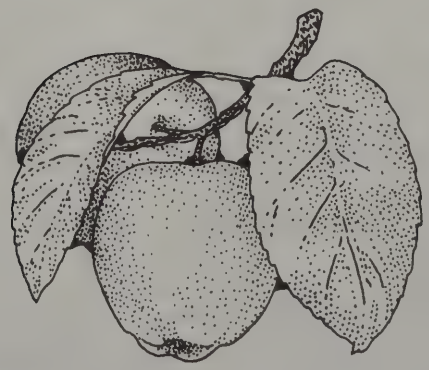
The disk contains two versions of BASIC. The *binary version* computes quickly: it's used by scientists who want blazing speed. The *decimal version* computes slowly but handles decimals with extra accuracy: it's used by accountants who are in no rush, but insist on never being off by a penny.

To choose the binary version, point at the icon above "Microsoft BASIC (b)". To choose the decimal version instead, point at the icon above "Microsoft BASIC (d)".

After you've pointed to the icon, double click. The icon will explode.

Near the bottom of the screen, you'll see the word "Command". Point at that word, and then click (just once). You'll see horizontal stripes next to the word.

Instead of saying OK, the computer shows a blinking vertical line.



ARITHMETIC

When you type a command (such as PRINT 5+2), the command appears at the bottom of the screen, in the window that's titled "Command". The computer prints the answer (7) at the top of the screen.

How to quit. Whenever you're done using BASIC, point at the word "File" (which is at the top of the screen). Hold down the mouse's button. While you hold down the mouse's button, you'll see this menu:

```
New
Open...
Close
Save
Save as...
Print...
Quit
```

While you keep holding down the mouse's button, point at the word "Quit"; then lift your finger off the button.

Then the computer might say:

```
Current program is not saved.
Do you want to save it?
```

If the computer says that, point at the word "No", then click.

The screen will grow dark, and you'll see "Microsoft BASIC" near the upper right corner.

Hold down the COMMAND key; and while you keep holding down the COMMAND key, tap the E key. The computer will eject the disk from the drive.

Then you can insert a different disk, or turn off the power.

LONG COMMANDS ON SMALL SCREENS

Keep your commands short. If you type a command that's too long to fit on a line, the computer has difficulty handling it.

HOW TO PROGRAM

Programming your computer consists of three steps. . . .

Step 1: say NEW. But instead of typing the word NEW, do the following. . . .

Point at the word "File" (which is at the top of the screen). Hold down the mouse's button. While you hold down the mouse's button, you'll see this menu:

```
New
Open...
Close
Save
Save as...
Print...
Quit
```

While you keep holding down the mouse's button, point at the word "New"; then lift your finger off the button. If you're lucky, you'll see horizontal stripes next to the word "List".

If you're unlucky, the computer will say:

```
Current program is not saved.
Do you want to save it before proceeding?
```

Point at the word "No", then click (once). You'll see horizontal stripes next to the word "List".

Step 2: type your program. Whatever you type will appear underneath the word "List", in the *list window*.

You do *not* have to type a number in front of each line; those numbers are optional. Instead of typing—

```
1 print "I love you"
2 print "you turn me on"
3 print "let's get married"
```

you can type:

```
print "I love you"
print "you turn me on"
print "let's get married"
```

Step 3: run the program. Instead of typing the word RUN, hold down the COMMAND key (which has the cloverleaf on it); and while you keep holding down the COMMAND key, tap the R key. (You do *not* have to press the ENTER key afterwards.)

The list window will disappear. The computer will print everything you requested; it will print:

```
I love you
you turn me on
let's get married
```

Then the list window will reappear, and you'll see your program in the window again.

Review. Here's the rule: if you type a list of instructions in the list window, the computer won't obey those instructions until you tap the R key while holding down the COMMAND key.

Changing the list window. If your program's too long to fit in the list window, the window will show just *part* of your program.

On the right side of the list window, you'll see an up-arrow and a down-arrow. If you point at the up-arrow and then hold down the mouse's button a while, you'll see the upper part of your program. If you point at the down-arrow and then hold down the mouse's button, you'll see the lower part of your program.

At the bottom of the list window, you'll see a left-arrow and a right-arrow. If you point at the left-arrow and then hold down the mouse's button, you'll see the left part of your program. If you point at the right-arrow and then hold down the mouse's button, you'll see the right part of your program.

To enlarge the list window, point at the word "List", then double click. The list window will expand, so that it consumes the entire screen.

To shrink the list window back to its original size, point at the word "List", then double click again.

To make the entire list window disappear, point at the small square that's left of the word "List", and click.

To make the entire list window reappear, hold down the COMMAND key; and while you keep holding down the COMMAND key, tap the L key.

MAKE A MISTAKE?

If you mess up a line, do *not* retype it underneath. Instead, do the following. . . .

Point at the right edge of the rightmost character that you want to revise. Click once. Press the BACKSPACE key several times, until all the bad characters disappear. Then type the characters you wish to insert. When you've finished typing them, point underneath the entire program, and click once.

What if the computer gripes? If you accidentally type PRIMT instead of PRINT, when you run the program the computer will gripe by saying:

```
Undefined subprogram
```

It will also put your program on the screen, and will draw a box around the line it griped about. Press the RETURN key, then revise the bad line.

HOW THE COMPUTER REARRANGES YOUR PROGRAM

The computer does *not* rearrange your program! The computer does *not* put the lines in numerical order! If you type 80, the computer will *not* erase line 80!

How to erase a line. Point slightly left of the leftmost character in the line—so that you're pointing at the gap between the leftmost character and the window's left wall. Press the mouse's button; while you keep pressing the button, move down one line, but without touching any characters. The line you want to erase will become black. Lift your finger off the button. Press the BACKSPACE key. The line will disappear. Point underneath the entire program, then click.

How to insert a new line. Point at the right edge of the rightmost character of line above. Click. Press the RETURN key. You'll see a gap, to put the new line in. Type the new line; do *not* press the RETURN key at the end of it. Point underneath the entire program. Click.

How to list. The computer tries to *always* show the list of your program on the screen. If the computer ever fails to show the list on the screen, do *not* type the word LIST: instead, tap the L key while holding down the COMMAND key.

RANGES OF LINES

To list from line 30 to the end of your program, point at the word "Command", click, and type:

```
list 30
```

The computer does *not* understand "LIST 30-80" or "LIST -80".

To delete lines 30 through 80, point at the word "Command", click, and type:

```
delete 30-80
```

At the end of that command, tap the ENTER key (or RETURN key). Then tap the L key while holding down the COMMAND key.

CLEARING THE SCREEN

Your keyboard lacks a CLEAR key.

HOW TO PRINT ON PAPER

Your keyboard lacks a PRINT key. Instead of pressing a PRINT key, press the CAPS LOCK key, so that it stays down; then hold down the SHIFT and COMMAND keys; while you keep holding down the SHIFT and COMMAND keys, tap the 4 key. The printer will print onto paper a copy of the entire screen. When the printer has finished, tap the CAPS LOCK key, so that it pops back up.

Your keyboard lacks an ECHO key.

Your computer understands LPRINT.

Your computer understands LLIST. But before you type LLIST, point at the word "Command" and click. Then type LLIST. The printer will list your program on paper. When the printer's finished, tap the L key while holding down the COMMAND key.

GO TO

Instead of typing this—

```
10 print "cat"
20 print "dog"
30 go to 10
```

you can type:

```
10 print "cat"
print "dog"
go to 10
```

If you delete the 10 from the top line, the computer will gripe when it sees "go to 10".

Instead of using a number (such as 10), you can use a name (such as JOE), but you must put a colon after the name, like this:

```
joe: print "cat"
print "dog"
go to joe
```

To abort a program, hold down the COMMAND key; and while you keep holding down the COMMAND key, tap the period key.

After you've aborted a program, if you wish to see a LIST, tap the L key while still holding down the COMMAND key.

How to trace your program. If a program runs faster than your eye can see or your mind can think, slow it down. To do that, instead of saying RUN (by tapping R with the COMMAND key), say TRACE (by tapping T with the COMMAND key).

That makes the computer do just one line or subline of your program; then it will pause. When you give the TRACE command again (by tapping T with the COMMAND key again), the computer will do the next line or subline. Each time you tap the T key while holding down the COMMAND key, the computer will do one more line or subline. (To make the computer to *several* lines or sublines, hold down the T key for a while, instead of tapping it.) Each time the computer does a line, it also shows the program in the list window, and puts a box around the line it just did.

Love. If a program involves a semicolon with a GO TO, begin the program by saying WIDTH 60, like this:

```
width 60
10 print "love;"
go to 10
```

If you forget to say WIDTH 60, the computer will print just one line of love and then do nothing until you abort.

STOP

While the computer is running a program, if it encounters the word STOP, it will say:

```
Program stopped.
```

Press the RETURN key. The computer will list your program in the list window. The computer will put a box around the line of your program that said STOP.

DISKS

If you buy a blank disk and want to use it, you must turn the blank disk into a Macintosh disk. Here's how. . . .

Turn the computer off, put the Microsoft BASIC disk into the drive, and turn the computer back on. Point at the Microsoft BASIC disk icon, and double click. The icon will explode.

While you hold down the COMMAND key, tap the E key. The computer will eject the Microsoft BASIC disk from the drive.

Put a blank disk into the drive. The computer will say:

```
This disk is unreadable.
Do you want to initialize it?
```

Point at the word "Initialize", and click. The computer will say, "Initializing disk."

After a few seconds, it will say, "Please name this disk." Invent a name for the disk; type the name, and press the RETURN key afterwards.

On the right side of the screen, you'll see an icon that stands for the disk, and underneath the icon you'll see the name you invented. You've turned the disk into a Macintosh disk.

But to use the disk easily, you should copy "Microsoft BASIC (b)" and "System Folder" onto that disk. Here's how. . . .

Point at the icon labeled "Microsoft BASIC (b)". [If you prefer the decimal version of Microsoft BASIC, point at the icon labeled "Microsoft BASIC (d)" instead.] Then click. The icon will become darker.

Point at the icon labeled "System Folder". Hold down the SHIFT key; and while you keep holding down the SHIFT key, click the mouse's button. The icon will become darker. So on your screen, you'll see two dark icons: "Microsoft BASIC (b)" and "System Folder". Then lift your finger off the SHIFT key.

Your next job is to drag the System Folder icon to your new disk's icon. To do that, point at the System Folder icon, hold down the mouse's button, and—while you keep holding down the mouse's button—point at your new disk's icon. Then lift your finger from the button.

The computer will eject the new disk and tell you to insert the Microsoft BASIC disk. Obey the computer: insert the Microsoft BASIC disk.

Five times, the computer will tell you to insert the new disk and then the Microsoft BASIC disk. Obey the computer: swap the disks five more pairs of times. Then the computer will stop printing messages about swapping disks. Your new disk has become an easy-to-use BASIC disk.

To test your new disk, eject the Microsoft BASIC disk (by tapping the E key while holding down the COMMAND key). Turn off the computer, insert your new disk, then turn the computer back on. If your disk works correctly, the upper right corner of the screen will show your disk's icon and its label. Point at the icon, double click, and you'll see three more icons. Point at the icon labeled "Microsoft BASIC", double click, and you should be in the middle of using BASIC.

Saving. You should *not* save programs on the original "Microsoft BASIC" disk. Instead, save your programs on a new disk that you turned into an easy-to-use BASIC disk.

So before you write programs to save, turn off the computer, insert a new easy-to-use BASIC disk, and then turn the computer back on.

How to give a disk command. Your computer understands all the standard disk commands. You can say SAVE "JOE" and FILES and LOAD "JOE" and RUN "JOE". But before you give one of those commands, point at the word "Command" and click.

When you've finished giving such commands, and want to type a program instead, tap the L key while holding down the COMMAND key.

Your program's name can be short (such as JOE) or long: up to 255 characters. Short names are better, because they can fit in small windows.

Instead of saying RENAME "JOE" TO "FRED", say NAME "JOE" AS "FRED".

TAPES

Your computer can't handle tapes.

EDITING

To insert some text between two characters, point at the gap between the two characters, click, then type the text.

To delete a group of characters, point at the left edge of the first character in the group, hold down the mouse's button, and—while still holding down the mouse's button—point at the right edge of the last character in the group. The characters you want to delete will turn black. Lift your finger off the mouse's button. Press the BACKSPACE key. The characters will disappear.

RENUMBERING

Your computer doesn't understand RENUM.

PAUSING

Your keyboard lacks a PAUSE key. To make the computer pause, tap the S key while holding down the COMMAND key.

To stop pausing, press the SPACE bar.

COUNTING TO 100

To indent, you can hit the SPACE bar repeatedly. But to indent more easily, hit the TAB key. Hitting the TAB key has the same effect as hitting the SPACE bar four times.

If you indent a line (by hitting the SPACE bar repeatedly, or by hitting the TAB key), the computer will assume you want to indent the line underneath also, and so the computer will automatically indent that line for you.

If the computer indents a line that you don't want indented, erase the indentation, by hitting the BACKSPACE key four times.

ZONES

Your screen has 4 zones. Zones 1-4 are each 14 characters wide; zone 5 is 18 characters wide. Altogether, the screen is 60 characters wide.

Those measurements are for standard-size characters. Your computer can also produce characters that are extra-narrow or extra-wide.

If a program involves zones with a GO TO, begin the program by saying WIDTH 60, like this:

```
width 60
10 print "hi"
go to 10
```

PIXELS

The X coordinate goes from 0 to 490. The Y coordinate goes from 0 to 253.

Fundamental shapes. Say PSET instead of PLOT. The computer draws each shape in black, on a white background.

Painting. Your computer doesn't understand the word PAINT.

Colors. The standard Macintosh displays just black and white: it can't display any other colors. The code number for white is 0; the code number for black is 1. (Any even number produces white; any odd number produces black.)

SOUNDS

Your computer doesn't understand the word PLAY.

RANDOM INTEGERS

Instead of saying just RANDOMIZE, say RANDOMIZE TIMER. To get a random integer from 1 to 5, instead of saying RND(5), say 1+INT(RND*5).

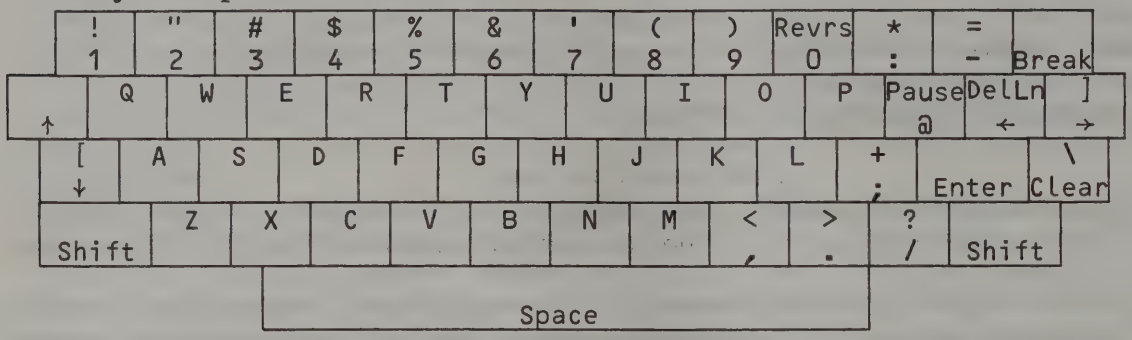
TRS-80 COLOR COMPUTER 2

CHOOSING A LANGUAGE

When you buy Radio Shack's TRS-80 Color Computer 2, make sure you get *extended* color BASIC, not *standard* color BASIC (which costs less but is crummy). The following comments apply to extended color BASIC.

THE KEYBOARD

The keys are placed like this:



The computer capitalizes all letters automatically; so to type a capital letter, do *not* press the SHIFT key.

There is no CAPS key.

The BACKSPACE key has a left-arrow on it; so if you tap the left-arrow key, the computer will erase the last character you typed. If you tap the left-arrow key *while holding down the SHIFT key*, the computer will erase the entire *line* you've been typing.

GET STARTED

Attach the computer to a TV. Turn on the computer's power, by pressing the POWER button, which is at the computer's rear. Turn on the TV, and turn it to channel 3 or 4. To switch the computer from channel 3 to channel 4 or back to channel 3, flip your computer's channel-selection switch, which is at the computer's rear. When the computer's on and functioning correctly, the computer will say OK.

RANGES OF LINES

Say DEL instead of DELETE.

HOW TO PRINT ON PAPER

Your keyboard doesn't have a PRINT key or an ECHO key. Instead of saying "LPRINT" say "PRINT#-2,". For example, to print "I LOVE YOU" on paper, type this:

```
PRINT#-2, "I LOVE YOU"
```

Even though the computer doesn't understand "LPRINT", it *does* understand "LLIST".

LONG VARIABLE NAMES

Variable names must be short. The name of a numeric variable must be just 1 or 2 characters: the name of a string variable must be just 1 or 2 characters followed by a dollar sign. If you try to make the variable's name longer than that, the computer will handle it unreliably.

If you want a variable name that's 2 characters long, avoid the following 2-character names, which have special meanings: GO, TO, IF, ON, FN, OR.

STRING INPUT

For some of the programs that involve string input, the computer will gripe about being "out of string space"; the computer will say:

```
?OS ERROR
```

To stop the computer from griping, insert this statement:

```
1 CLEAR 1000
```

If the computer *still* says OS ERROR, raise 1000 to a higher number, like this:

```
1 CLEAR 2000
```

Keep raising, until the computer stops that gripe. On the other hand, if you raise *too* high, the computer will say—

```
?OM ERROR
```

That means you must go lower.

So if the computer says "OS ERROR", you should raise the CLEAR number; if the computer says "OM ERROR", you should lower the CLEAR number.

Here's why. . . .

Your program contains two kinds of strings: the strings that are known, and the strings that are unknown. For example, if you say—

```
10 G$="DOWN"
```

the G$ is known to be DOWN. If you say—

```
10 INPUT "WHAT IS YOUR NAME";N$
```

the N$ is unknown, because its value depends on what the human inputs.

Normally, the computer reserves enough memory space to hold just 200 unknown characters. If your program needs *more* characters than that—for example, if your program inputs 7 strings, each having 30 characters, so that the total number of characters is 210—the computer will gripe, and say "out of string space error" (OS ERROR).

If you say—

```
1 CLEAR 1000
```

the computer will reserve enough space to hold 1000 unknown characters, instead of the normal quantity (which is just 200).

How much space can you reserve? That depends on how much memory you bought for your computer. If your computer has little memory, and you try to reserve more space than your computer has, the computer will give up, and say "out of memory error" (OM ERROR).

DISKS

When you flip the disk drive's power switch, you create an electrical surge that damages any disk inside the drive. So before flipping the drive's power switch, make sure the drive is empty.

When you're done using the computer, remove any disk from the drive, before you flip off the power.

How to insert a disk. To put a disk into the drive, open the drive's door, then insert the disk into the drive. When you insert it, make sure the disk's label is on top of the disk, and make sure the disk's big oval cutout goes into the drive *before* the label does. Close the drive's door.

(Exception: if your drive was made before 1985, the drive's slit is vertical instead of horizontal, and the disk's label is on the left side of the disk instead of the top.)

How to turn a blank disk into a Radio Shack disk. If you buy a blank disk and want your computer to use it, you must turn the blank disk into a Radio Shack disk.

To do that, insert the blank disk into the drive (and close the door). Then type DSKINI0. (That last character is a zero, not the letter O.)

The disk drive's red light will glow for 40 seconds. When the light stops glowing, the computer will say OK.

How to choose a name. Your program's name can be short (such as JOE) or long: up to 8 characters. The name's first character must be a letter; the remaining characters should be letters or digits.

Differences. Instead of saying FILES, say DIR (which stands for "directory").

Instead of saying KILL "JOE", say KILL "JOE/BAS", to emphasize that JOE was written in BASIC. Instead of saying RENAME "JOE" TO "FRED", say RENAME "JOE/BAS" TO "FRED/BAS". You must add the /BAS when you say KILL or RENAME, but not when you say SAVE or LOAD or RUN.

How to use drive 1. The main drive is called *drive 0*. If you have an extra drive, that extra is called *drive 1*.

Normally, the computer uses drive 0. If you want JOE to be in drive 1 instead of drive 0, put a colon and a 1 after JOE, like this:

```
SAVE "JOE:1"
```

To make the computer print the names of all the programs in drive 1, type this:

```
DIR 1
```

TAPES

To copy a program onto a tape, position the tape and notice the counter's number. Then press the recorder's RECORD and PLAY buttons at the same time as each other. Type this:

```
CSAVE "A"
```

The computer will copy the program onto tape and then say OK. Press the recorder's STOP button.

How to read a program from a tape. Put the tape into the recorder. (To do that, you might have to press the recorder's EJECT button.)

Type the word CLOAD.

Press the recorder's REWIND button. When the tape is rewound, press the recorder's STOP button. Push the counter's button, so the counter becomes zero.

Press the recorder's FAST FORWARD button. Look at the counter. When the number on the counter is almost up to the number of the program's beginning, press the recorder's STOP button.

Press the recorder's PLAY button. The computer will search on the tape for the beginning of the program. When the computer reaches the beginning of the program, the computer will print an F.

Then the computer begins reading the program. When the tomputer finishes reading it, the computer will say OK.

Press the recorder's STOP button. Type the word LIST or RUN.

EDITING

To edit line 20, type:

```
EDIT 20
```

If the first character in line 20 is correct (and doesn't need to be edited), press the SPACE bar. If the second character in line 20 is correct also, press the SPACE bar again. Press the SPACE bar for each correct character. Each time you press the SPACE bar, the character that you said was correct appears on the screen.

When you get to the point where the next character on the screen would be *in*correct, press the D key instead of the SPACE bar. The D key means: delete the next character. Pressing the D key prevents the next character from appearing on the screen.

To insert extra characters in the line, press the SPACE bar several times, until you get to where you want the characters to be inserted. Then press the I key (which tells the computer you want to insert). Then type the characters you want to insert. After typing them, tap the up-arrow key while holding down the SHIFT key.

When you've finished all the deletions and insertions for that line, press the ENTER key. That makes the screen show the remainder of the line (the right-hand part of the line), and tells the computer you've finished editing the line.

PAUSING

To make the computer pause, hold down the SHIFT key; and while you keep holding down the SHIFT key, tap the @ key.

To stop pausing, press the SPACE bar.

COUNTING TO 100

If you indent the lines between FOR and NEXT, your computer ignores the indentation. When you LIST your program, you won't see any indentation. So don't bother to indent.

ZONES

Each zone is 16 characters wide. Your screen has just 2 zones, so your screen is 32 characters wide.

TAB

On other computers, the leftmost character in the line is said to be at position 1. But on your computer, the leftmost character in the line is said to be at *position 0* instead; after position 0 come positions 1, 2, 3, etc.

So on your computer, if you say TAB(0), you'll be going to the leftmost position; if you say TAB(1), you will *not* be going to the leftmost position.

LOCATE

Your computer doesn't understand the word LOCATE. Here's what to do instead . . .

Your computer considers the screen's top line to be *row 0*, and the line below it to be *row 1*. Your computer considers the row's leftmost character to be in *column 0*, and the next character to be in *column 1*.

To print the word DROWN, so that it begins at row 3 and column 7, type this:

```
PRINT@32* 3+7, "DROWN"
```

To use pixels, type these lines:

```
10 PMODE 4: PCLS: SCREEN 1
10000 GO TO 10000
```

Line 10 warns the computer that you want to use pixels, and arranges the screen so that the X coordinate can go from 0 to 255, and the Y coordinate can go from 0 to 191. Line 10000, at the bottom of your program, makes the computer pause, so that you can admire whatever graphics have been drawn. Between lines 10 and 10000, you can insert lines containing the following commands. . . .

To draw a dot at pixel (100,100), say PSET (100,100).

To draw a line from pixel (0,0) to pixel (100,100), say LINE (0,0)-(100,100),PSET.

To draw a box ("rectangle") that has a corner at pixel (0,0) and has an opposite corner at pixel (100,100), say LINE (0,0)-(100,100),PSET,B.

To draw a box that has a corner at (0,0) and an opposite corner at (100,100), and also make the computer fill in the box (i.e., paint the box's interior), say LINE (0,0)-(100,100),PSET,BF.

Each of those commands contains the word PSET. To erase the shape that you drew, give the same command again, but replace the PSET by PRESET.

To draw a circle centered at pixel (100,100), and to make the circle's radius be 50, say CIRCLE (100,100),50. To erase that circle, say CIRCLE (100,100),50,0.

To paint the inside of any shape (such as a circle or a person), say PAINT and mention a point inside the shape. For example, to fill in an outline that has (101,101) inside it, say PAINT (101,101). To erase that paint (and also the shape itself), say PAINT (101,101),0.

To erase the entire drawing ("clear the screen"), say PCLS.

If you say SCREEN 0, the screen will stop showing graphics (and will show text instead), until you say SCREEN 1 (which brings back the graphics).

To get started experimenting with all those commands, run the following program; it draws a line from (0,0) to (100,100) and then draws a circle centered at pixel (100,100) and with radius 50:

```
10 PMODE 4: PCLS: SCREEN 1
20 LINE (0,0)-(100,100)
30 CIRCLE (100,100),50
10000 GO TO 10000
```

The computer draws the shapes in green, on a black background. After you've admired the pretty picture, abort the program, by pressing the BREAK key.

Colors. To get colors, line 10 must say PMODE 3 instead of PMODE 4.

After you've said PMODE 3, whenever the computer draws (or erases) a pixel, it does the same thing to the pixel next to it. Specifically, if you tell the computer to draw (or erase) a pixel whose X coordinate is even, the computer will do the same thing to the pixel at its right (whose X coordinate is odd). If you tell the computer to draw (or erase) a pixel whose X coordinate is odd, the computer will do the same thing to the pixel at its left (whose X coordinate is even).

After you've said PMODE 3, the computer normally draws red shapes on a green background. But if you don't like those colors, you can pick different colors instead.

To choose different colors, use these code numbers: 1=green, 2=yellow, 3=blue, and 4=red. For example, to draw yellow shapes on a blue background, say COLOR 2,3. The traditional place to say COLOR 2,3 is in line 10, like this:

```
10 PMODE 3: COLOR 2,3: PCLS: SCREEN 1
```

If you prefer, you can say COLOR 2,3 later in your program instead. If you say it later, the shapes at the beginning of your program will be unaffected by that COLOR statement. When you draw new shapes after giving that COLOR statement, the shapes that were already on the screen (in their old colors) remain unchanged; they keep their original colors.

When you say PRESET (in the middle of a dot or line or box command), the computer assumes you want the shape erased; to erase the shape, the computer redraws the shape by using the background color. If you recently changed the background color (by saying COLOR), the erasure will be done by using the new background color, which won't match the old background color.

If you say CIRCLE (100,100),50,2 the computer will draw the circle in yellow (because the 2 at the end of the CIRCLE command is the code number for yellow). Do *not* put a 0 at the end of the CIRCLE command, since 0 is not an acceptable color number. You must pick a number from 1 to 4.

If you say PAINT, you must say which pixel to start painting at; you must say which color paint to use; and you must remind the computer of what color the outline had that you're filling in. For example, if you want the computer to paint starting at pixel (101,101)—which is inside the outline— and you want the computer to use yellow paint (whose code number is 2), and the outline that you're filling in was drawn in blue (whose code number is 3), say PAINT (101,101),2,3.

Near the end of your program, if you say SCREEN 1,1, you'll see a "negative" of your picture. In the "negative", green becomes light blue (even though Radio Shack's manual incorrectly says it becomes "buff"), yellow becomes cyan (greenish blue), blue becomes magenta (purplish red), and red becomes orange. All future drawings (even by other programs) will be done using those "negative" colors, unless you tell the computer to return to a "positive", by saying SCREEN 1,0.

SOUNDS

Your computer doesn't understand the word BEEP.

If you type—

```
SOUND 100,15
```

the computer will produce a sound whose pitch is 100. You can choose any pitch from 1 to 255.

In that example, the 15 makes the computer produce the sound for 1 second. If you want the sound to last longer—so that it lasts 2 seconds—replace the 15 by 15*2. For 10 seconds, say 15*10. (That's because your computer's metronome beats 15 times per second.)

PLAY. Your computer handles the PLAY command slightly differently than other computers.

Your computer can't play octaves 0 or 6: it plays just octaves 1 through 5. If you don't specify an octave, the computer will play in octave 2.

If you say PLAY "T1", the computer will play 56 quarter notes per minute, which is very slow (largo). If you say PLAY "T2", the computer will play twice as fast: 112 quarter notes per minute (moderato). If you say PLAY "T3", the computer will play three times as fast: 168 quarter notes per minute (vivace). You can pick speeds up to PLAY "T255", which is ridiculously fast! If you don't specify a tempo, the computer will play at speed T2, which is 112 quarter notes per minute (moderato).

Normally, the computer plays at volume 15; but you can change the volume. To switch to volume 4 (which is lower, quieter), say:

```
PLAY "V4"
```

You can switch to any volume from 1 to 31.

If you try the bottom example in the main text, slow down the tempo: change T150 to T2.

RANDOM INTEGERS

Omit the word RANDOMIZE.

RANDOM DECIMALS

For a random decimal between 0 and 1, say RND(0) instead of RND.

TRS-80 MODEL 4

CHOOSING A LANGUAGE

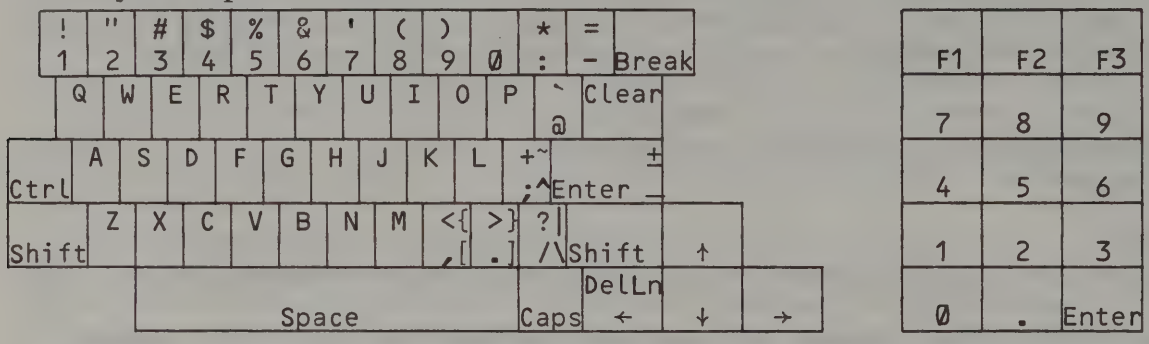
Radio Shack's TRS-80 Model 4 computer can handle two versions of BASIC. The best version, called *Model 4 BASIC*, requires a disk called *Model 4 TRSDOS and BASIC Interpreter*, and requires a drive to put it in. The alternate version, called *Model 3 BASIC*, is only for people who neglected to buy a disk drive, or who lost the disk called "Model 4 TRSDOS and BASIC Interpreter", or who want to run obsolete "Model 3" software.

I'll assume you have the disk called "Model 4 TRSDOS and BASIC Interpreter" and a drive to put it in, so that you can run Model 4 BASIC. (If you don't have that disk and drive, buy them now!)

Here's how to use Model 4 BASIC. . . .

THE KEYBOARD

The keys are placed like this:



Exception: if you bought the computer before 1984, the Ctrl key, CLEAR key, and arrow keys are located differently.

The zero key is Swedish.

In that diagram, one of the keys looks like this:

If you tap that key, you'll be typing a comma. If you tap that key *while holding down the SHIFT key*, you'll be typing the symbol "<". If you tap that key *while holding down the CLEAR key*, you'll be typing the symbol "[". To type the remaining symbol on that key (the "{"), do this: hold down the CLEAR key; and while you keep holding down the CLEAR key, hold down the SHIFT key; and while you keep holding down the CLEAR and SHIFT keys, tap the key that shows the "{".

The BACKSPACE key has a left-arrow on it; so if you tap the left-arrow key, the computer will erase the last character you typed. If you tap the left-arrow key *while holding down the SHIFT key*, the computer will erase the entire *line* you've been typing.

GET STARTED

When you flip the power switch, you create an electrical surge that damages any disks inside the computer. So before flipping the power switch, remove any disks from the computer.

Then flip on the computer's power switch, which is underneath the computer's right side. The red light on the bottom disk drive will turn on, then off.

Insert a disk that contains "Model 4 TRSDOS and BASIC Interpreter" into the bottom drive; when you insert it, make sure the disk's label is on *top* of the disk, and make sure the disk's big oval cutout goes into the drive *before* the label does. Close the drive's door.

On the far right side of the keyboard, you'll see an orange button. Press it.

The computer will say "Date". If the writing looks fuzzy, turn the brightness and contrast dials, which are underneath the computer's left side. Type the date. For example, if today is January 27, 1986, type 01/27/86 and then press the ENTER key.

The computer will say:

```
TRSDOS Ready
```

Type this word:

```
basic
```

Then press the ENTER key. After a brief pause, the computer will say:

```
Ready
```

Notice that the computer says "Ready" instead of "OK".

Whenever you're done using the computer, remove any disks from the drives, before you flip off the power.

MAKE A MISTAKE?

If the computer gripes at you, press the ENTER key immediately.

After pressing the ENTER key, correct the error, by retyping the line that the computer griped about.

CLEARING THE SCREEN

To clear the screen by using the CLEAR key, tap the CLEAR key *while holding down the SHIFT key*.

HOW TO PRINT ON PAPER

Your keyboard lacks a PRINT key. Instead of pressing a PRINT key, tap the colon key (which is next to the zero key) *while holding down the CTRL key*.

Your keyboard lacks an ECHO key. Instead of pressing an ECHO key, type this:

```
system "link *do *pr"
```

To stop the echo, type this:

```
system "reset *do"
```

DISKS

If you buy a blank disk and want your computer to use it, you must turn the blank disk into a Radio Shack disk.

To do that, put the blank disk into the top drive. Type:

```
system
format :1 (q=n)
```

The computer will turn the blank disk into a Radio Shack disk. Then the computer will say "TRSDOS Ready".

Put Model 4 BASIC onto that disk, by typing:

```
backup :0 :1
```

The computer will ask whether you're sure; type the letter y. The computer will copy all information from the bottom disk to the top disk. Then the computer will say "TRSDOS Ready" again.

Tell the computer you want to use BASIC now, by typing:

```
basic
```

The computer will say "Ready".

Directory of files. Instead of typing the word FILES, type this:

```
system "dir"
```

That makes the computer print a directory, showing the names of all the programs in all your disk drives.

How to choose a name. Your program's name can be short (such as JOE) or long: up to 8 characters. The name's first character must be a letter; the remaining characters should be letters or digits.

How to use drive 1. The main drive is called *drive 0*. The other drive is called *drive 1*.

Normally, the computer uses drive 0. If you want JOE to be in drive 1 instead of drive 0, put a colon and a 1 after JOE, like this:

```
save "joe:1"
```

TAPES

Model 4 BASIC doesn't let you use tapes. Since you have disks, which are better than tapes, you wouldn't want to use tapes anyway.

EDITING

To edit line 20, type:

```
edit 20
```

If the first character in line 20 is correct (and doesn't need to be edited), press the SPACE bar. If the second character in line 20 is correct also, press the SPACE bar again. Press the SPACE bar for each correct character. Each time you press the SPACE bar, the character that you said was correct appears on the screen.

When you get to the point where the next character on the screen would be *in*correct, press the D key instead of the SPACE bar. The D key means: delete the next character. Pressing the D key causes the next character to be surrounded by backslashes, which mean the character is being deleted.

To insert extra characters in the line, press the SPACE bar several times, until you get to where you want the characters to be inserted. Then press the I key (which tells the computer you want to insert). Then type the characters you want to insert. After typing them, tap the up-arrow key while holding down the SHIFT key.

When you've finished all the deletions and insertions for that line, press the ENTER key. That makes the screen show the remainder of the line (the right-hand part of the line), and tells the computer you've finished editing the line.

PAUSING

Your keyboard lacks a PAUSE key.

To make the computer pause, tap the @ key while holding down the SHIFT key. (Normally, pressing those keys makes the screen display the symbol "`"; but if you press those keys *while a program is running*, the computer will pause instead.)

To stop pausing, press the SPACE bar.

COUNTING TO 100

To indent you can hit the SPACE bar repeatedly. But to indent more easily, hit the right-arrow key once.

LOCATE

Your computer doesn't understand the word LOCATE. Here's what to do instead . . .

Your computer considers the screen's top line to be *row 0*, and the line below it to be *row 1*. Your computer considers the row's leftmost character to be in *column 0*, and the next character to be in *column 1*.

To print the word DROWN, so that it begins at row 3 and column 7, type this:

```
print@(3,7), "drown"
```

PIXELS

To make Model 4 BASIC handle pixels, you must buy a graphics board, which costs more than most people can afford. If you're rich enough to buy it, read the instructions that come with it.

SOUNDS

Your computer doesn't understand the words BEEP and PLAY.

If you type—

```
sound 4,10
```

the computer will produce a sound whose pitch is 4 and whose duration is 10. The pitch can be any integer from 0 to 7: 0=D#,1=E, 2=F, 3=F#, 4=G, 5=A, 6=A#, 7=B. The duration can be any integer from 0 to 31: 0 means half a second; 31 means about 15 seconds.

RANDOM INTEGERS

Say RANDOM instead of RANDOMIZE.

RANDOM DECIMALS

For a random decimal between 0 and 1, say RND(0) instead of just RND.

OTHER COMPUTERS

IBM

IBM's most popular computer is the *IBM Personal Computer (IBM PC)*, which I explained several pages ago.

IBM's other popular microcomputers are the *IBM PC Junior*, the *IBM Portable PC*, the *IBM PC XT*, and the *IBM PC AT*. They all resemble the IBM PC. So if you have one of those computers, read my explanation of the IBM PC.

All those computers typically come with disk drives. The main disk drive is called *drive A*; that's the drive into which you put the DOS system master disk. Other disk drives are called *drive B* or *drive C*. Here's where they are:

Computer	Where the drives are
IBM PC	Drive A is on the left. Drive B is on the right.
IBM PC Junior	Drive A is inside the computer. Drive B is separate.
IBM Portable PC	Drive A is on top of drive B.
IBM PC XT	Drive A is on the left. Drive C is on the right.
IBM PC AT	Drive A is top left. B is bottom left. C is at the right.

Junior. If you have an IBM PC Junior, you should buy the *BASIC cartridge* and put it into the left cartridge slot, before you turn the computer on. (If you *don't* buy the BASIC cartridge, your computer won't understand the words CIRCLE, PAINT, PLAY, and TIMER, and won't understand the BASIC words for handling disks.)

The Junior's keyboard is different:

Walter keyboard	IBM PC keyboard	IBM PC Junior keyboard
Type RUN.	Press the F2 key.	Press the Fn key then the F2 key.
Type LIST.	Press the F1 key.	Press the Fn key then the F1 key.
Press CLEAR key.	HOME with Ctrl.	While pressing Ctrl, tap Fn then HOME.
Press PRINT key.	PrtSc with SHIFT.	Press the Fn key then the PrtSc key.
Press ECHO key.	PrtSc with Ctrl.	Press the Fn key then the ECHO key.
Press BREAK key.	BREAK with Ctrl.	Press the Fn key then the BREAK key.
Press PAUSE key.	Num Lock with Ctrl.	Press the Fn key then the PAUSE key.

If your Junior's attached to a monitor instead of a TV, type:

```
width 80
```

To increase your Junior's abilities, instead of saying SCREEN 1, say:

```
clear,,,32768: screen 5
```

That lets your Junior handle all 16 colors simultaneously, instead of just 4 colors. The colors are numbered from 0 to 15. For example, to draw a line from pixel (0,0) to pixel (100,100) using color 14 (which is yellow), and then draw a yellow circle of radius 50 there, say:

```
clear,,,32768: screen 5
line (0,0)-(100,100),14
circle (100,100),50,14
```

If you turn up the volume on your TV or monitor, and then say SOUND ON, you can make your Junior's three voices sing simultaneously, like this:

```
play "gab","efg","ccd"
```

While the first voice is singing "gab", the second voice sings "efg", and the third voice sings "ccd".

IMITATIONS OF THE IBM PC

Imitations of the IBM PC have been made by many companies (such as Compaq, Zenith, AT&T, ITT, Panasonic, Leading Edge, NCR, Sperry, Columbia, Corona, Sanyo, and Seequa). Those imitations are called *IBM PC clones*.

If you have an IBM PC clone and want to use BASIC, insert the MS-DOS system master disk before you turn on the computer. Say BASIC instead of BASICA.

COMMODORE

Commodore's most popular computer is the *Commodore 64*, which I explained several pages ago.

Commodore's other low-cost computers are the *Commodore 16*, the *Commodore 128*, the *Commodore Plus 4*, the *SX-64*, the *Vic-20*, the *Pet*, the *Super Pet*, and the *CBM*. They all resemble the Commodore 64. So if you have one of those computers, read my explanation of the Commodore 64.

Commodore also makes a more expensive computer, called the *Amiga*, which resembles the Apple Macintosh. So if you have an Amiga, read my explanation of the Apple Macintosh.

Some Commodore computers use different commands for handling disks:

Most computers	64, SX-64, Vic	16, Plus 4, Pet, Super Pet, CBM
SAVE "JOE"	SAVE "JOE",8	DSAVE "JOE"
LOAD "JOE"	LOAD "JOE",8	DLOAD "JOE"
RUN "JOE"	LOAD"JOE",8 RUN	DLOAD "JOE" RUN
FILES	LOAD"JOE",8 RUN	*16, SX-64, Plus 4:* DIRECTORY *Pet, Super Pet, CBM:* CATALOG DO
DELETE "JOE"	OPEN 15,8,15,"S:JOE" CLOSE 15	SCRATCH "JOE"
RENAME "JO" TO "AL"	OPEN 15,8,15,"R:AL=JO" CLOSE 15	RENAME "JO" TO "AL"
FORMAT	OPEN 15,8,15,"N:ELIZ,OO" CLOSE 15	HEADER "ELIZ",IOO,DO

When you start using a Commodore 128, it resembles a Commodore Plus 4, until you say—

```
GO 64
```

which makes it imitate a Commodore 64 instead.

If you have a Vic-20, you must keep the INPUT statement's prompt short: no longer than 20 characters.

If you have a Pet or Super Pet or CBM, the CTRL and RESTORE keys are missing: instead of holding down the CTRL key, hold down the RVS key; instead of holding down the RUN STOP and RESTORE keys simultaneously, tap the RETURN key.

If you have a Pet or Commodore 16 or Commodore Plus 4, say R=RND(-TI) instead of RANDOMIZE.

ATARI

Atari's most popular computer is the *Atari 800 XL*, which I explained several pages ago.

Atari has also built several slight variations of the Atari 800 XL. They're called the *Atari 600 XL*, the *Atari 1200 XL*, the *Atari 400*, the *Atari 800*, and the *Atari 130 XE*. If you have one of those computers, read my explanation of the Atari 800 XL. Unfortunately, the Atari 400 and Atari 800 can*not* handle "GRAPHICS 15", and they require you to insert a cartridge even for non-Microsoft BASIC; to insert the cartridge, open the door on top of the computer, put the cartridge into the slot, and then close the door.

Atari also makes a more expensive computer, called the *Atari 520 ST*, which resembles the Apple Macintosh. So if you have an Atari 520 ST, read my explanation of the Apple Macintosh.

APPLE

Apple's most modern computers are the *Apple Macintosh* and the *Apple 2c*. I explained those computers several pages ago.

The *Macintosh XL* and the *Lisa* resemble the Apple Macintosh. So if you have one of those computers, read my explanation of the Apple Macintosh.

The *Apple 2*, *Apple 2+*, and *Apple 2e* all resemble the Apple 2c. So if you have one of those computers, read my explanation of the Apple 2c. Here are some additional comments about those computers. . . .

The keyboard. If you have a 2e, your keyboard lacks a 40 key and a KEYBOARD key. If you have a 2 or 2+, the zero key is Swedish, the CONTROL key says CTRL on it, the letters are automatically capitalized, the keys are rearranged slightly, and the following keys are missing: DELETE, TAB, CAPS LOCK, OPEN APPLE, SOLID APPLE, KEYBOARD, 40, down-arrow, up-arrow, left-bracket, right-bracket, backslash, and accent.

Get started. The disk drive and its slit are in a separate box (instead of being built into the right side of the computer). If you have *two* disk drives, begin by using disk drive #1, not disk drive #2.

Open the disk drive's door, by pulling the door out and up. Remove any disk from the drive. If you have a 2e, insert the "ProDOS User's Disk"; if you have a 2 or 2+, insert the "DOS 3.3 System Master Disk" instead. When inserting the disk, make sure the disk's label is on *top* of the disk, and make sure the disk's big oval cutout goes into the drive *before* the label does.

Close the door, by pushing it down and in.

Attach the computer to a TV or monitor. Turn on the TV or monitor. If you're using a TV, turn it on to channel 33.

On the back of the computer, you'll see the computer's power switch, which is black. Flip it to the ON position.

Here's what happens next, *if you have a 2e*:

At the top of the screen, a message will briefly appear. If your 2e is old (manufactured before February 1985), the message says "Apple]["; if your 2e is new (or you've upgraded your old 2e), the message says "Apple 2e". Look at the message, to double-check whether your 2e is old or new.

If your 2e is old, it can't handle lower-case letters well, so you must type just capitals. To type capitals easily, press the CAPS LOCK key, and make sure the CAPS LOCK key stays down.

Your 2e will also print this menu:

```
YOUR OPTIONS ARE:
     ? - TUTOR: PRODOS EXPLANATION
     F - PRODOS FILER (UTILITIES)
     C - DOS <-> PRODOS CONVERSION
     S - DISPLAY SLOT ASSIGNMENTS
     T - DISPLAY/SET TIME
     B - APPLESOFT BASIC
PLEASE SELECT ONE OF THE ABOVE
```

Tap the B key.

Regardless of whether you have a 2, 2+, or 2e, you'll eventually see this symbol near the left edge of the screen:

```
]
```

That symbol's called a *bracket*. It means everything's OK. Your computer prints a bracket instead of the word OK.

The characters are fat (40 per line). Your Apple might be able to make the characters thin (80 per line), but your Apple handles thin characters reliably only if your Apple is a *new* 2e that also contains an 80-column card. If you have that kind of Apple and want thin characters, say—

```
pr#3
```

and press the RETURN key.

Disks, if you have a 2e. To turn a blank disk into an Apple disk, turn the computer off, make sure drive 1 contains the ProDOS User's Disk, close the door, and turn the computer back on. The computer will say, "PRODOS USER'S DISK". Tap the F key; the computer will say, "FILER". Tap the V key; the computer will say, "VOLUME COMMANDS". Tap the F key; put the blank disk into drive 1; press the RETURN key three times; the computer will say, "FORMATTING". 25 seconds later, the computer will say, "FORMAT COMPLETE"; the blank disk has been turned into an Apple disk. Press the ESC key twice, then the Q key, then the RETURN key.

To make your 2e copy an entire disk, put the ProDOS User's Disk into drive 1. Turn the computer off, then on again. The computer will say, "PRODOS USER'S DISK". Tap the F key; the top of the screen will say, "FILER". Tap the V key; the top of the screen will say, "VOLUME COMMANDS". Tap the C key; take the disk you want to copy, and put it in drive 1; put a blank disk in drive 2; press the RETURN key six times. When the computer finishes copying, press the ESC key twice, then the Q key, then the RETURN key.

Disks, if you have a 2 or 2+. To turn a blank disk into an Apple disk, put the blank disk into drive 1 and then type:

```
NEW
10 PRINT "HELLO"
INIT HELLO
```

The drive's red light will glow for about 30 seconds. When it stops glowing, your blank disk is finally an Apple disk; it's called a "slave of a DOS 3.3 System Master". It's also a quick-start disk.

Your Apple 2 or 2+ doesn't understand CAT: say CATALOG.

To make your Apple 2 or 2+ copy an entire disk, put the DOS 3.3 System Master disk into drive 1. Type RUN COPYA and press the RETURN key. The computer will say, "APPLE DISKETTE DUPLICATION PROGRAM". Take the disk you want to copy, and put it in drive 1; put a blank disk into drive 2; press the RETURN key five times. The computer will copy all information from drive 1's disk to the blank disk. (In the process, it will also turn the blank disk into an Apple disk.) Then the computer will ask, "DO YOU WANT TO MAKE ANOTHER COPY?" If you don't, type an N and then press the RETURN key.

Editing. On the Apple 2 and 2+, the method of editing is complicated enough so that you shouldn't bother: just type the whole line over again. On the Apple 2e, the editing process is the same as for the 2c, with just two exceptions:

1. To begin the whole editing process, tap the ESC key, *then the left-arrow key*, then move the cursor up to the line you want to edit.

2. You see the cross only if your Apple 2e contains an 80-column card and you previously said PR#3. Otherwise, the cross is invisible: when the cross is at the cursor, the cursor covers up the cross, so that you can't see the cross (even though the cross is there).

FRANKLIN

Franklin manufactures the *Ace* computer, which comes in several versions. Most resemble the Apple 2+.

COLECO

The *Adam* computer, manufactured by Coleco, resembles the Apple 2+. Here are the major differences.

The keyboard. You can type either capital letters or lower-case letters.

Get started. The typical Adam uses tape (instead of disks). When you flip the power switch, you create an electrical surge that damages any tape inside the computer. So before flipping the power switch, remove any tape from the computer. Then flip on the computer's power switch.

Next, put the BASIC tape into the computer. Pull the COMPUTER RESET switch toward you. The tape will spin quickly, then slowly, and finally you should see the word BASIC at the top of the screen. (If your screen shows an electronic typewriter instead of the word BASIC, your BASIC tape is defective, and you must buy a new tape.)

Then remove the BASIC tape from the computer.

Arithmetic. To make sure BASIC is working properly, type this:

```
print 345+2
```

When you press the RETURN key at the end of that line, the computer should say:

```
347
```

(If the screen shows 47 or 7 instead of 347, the 3 is left of your screen's left edge: move the 3 farther to the right, by turning your TV's HORIZONTAL HOLD dial; if the HORIZONTAL HOLD dial doesn't work or is missing, try a different TV. If the second TV doesn't work either, try a different Adam.)

The Adam has trouble handling ".00". For example, if you type 3.00, the Adam will assume (incorrectly) that you mean a number slightly less than 3. If you type—

```
print 3-3.00
```

the Adam will print:

```
9.31322574E-10
```

All other computers would print 0.

The Adam has trouble handling numbers such as 3.00, 4.00, and 5.00. In fact, it has trouble handling every such number from 1.00 to 1024.00. (But it can handle 1025.00 correctly.)

Long commands on small screens. The Adam requires that each statement be short: no more than 127 characters. (Most other computers permit 255 characters instead.)

Clearing the screen. Although the Adam has a CLEAR key, that key doesn't work when you're using BASIC. Instead, hold down the CONTROL key; and while you keep holding down the CONTROL key, tap the L key.

How to print on paper. Although the Adam has a PRINT key, that key doesn't work when you're using BASIC. Instead, hold down the CONTROL key; and while you keep holding down the CONTROL key, tap the P key.

GO TO. Do *not* put a space between GO and TO: the space confuses the Adam. Say "GOTO", not "GO TO".

Disks & tapes. You can store programs on Adam's data tape. (Do *not* store programs on Adam's BASIC tape.) The commands for handling Adam's blank tape resemble the commands for handling Apple's disk. For example, to put a program onto the tape and name it JOE, say SAVE JOE.

But if you later revise JOE and put the revised version of JOE onto the tape (by saying SAVE JOE again), your tape will contain *two* versions of JOE: the revised version is called "A JOE"; the previous version is called "a JOE".

If you say LOAD JOE or RUN JOE or DELETE JOE, Adam assumes you're talking about the revised version ("A JOE"). To use the previous version instead of the revised version, say DELETE JOE (which deletes the revised version) and then say RECOVER JOE (which changes "a JOE" to "A JOE").

If your data tape ever acts strangely, it's probably because some of the information on it got wrecked flipping the power switch. The only way to use that tape again is to erase it completely. To erase the tape, type this:

```
init hello
```

That command erases everything on the tape, and reformats the tape so that it acts properly.

Zones. Your screen has just 2 zones. The left zone is 16 characters wide; the right zone is 15 characters wide. So the entire screen is 31 characters wide.

Pixels. If you draw a horizontal line (by using HPLOT) and then try to change the color of one of the points on the line, seven nearby points on the line will accidentally change color also.

After doing graphics, if you say TEXT, Adam will automatically blank the screen.

Random integers. Adam has trouble handling random numbers. If you run a program that contains random numbers and then run the same program again, you'll get the same "random" numbers as before.

TANDY'S RADIO SHACK

The Radio Shack division of the Tandy Corporation has sold a wide variety of computers.

Several pages ago, I explained two of Tandy's Radio Shack computers: the TRS-80 Color Computer 2 and the TRS-80 Model 4. Tandy's other computers fall into several categories. . . .

Computers that resemble the TRS-80 Model 4. If you have a TRS-80 Model 4P or 4D, read my explanation of the TRS-80 Model 4. (In the Model 4P, the disk drive's slit is vertical instead of horizontal; when putting a disk into that drive, make sure the disk's label is on the left side of the disk.)

Computers that resemble the TRS-80 Color Computer 2. If you have a TRS-80 Color Computer 1 or Micro Color Computer, read my explanation of the TRS-80 Color Computer 2. (If you have a Micro Color Computer, the power switch is on the right side, the channel-selection switch is underneath the computer, and you face several limitations: you can't use disks or standard pixel commands or random decimals, the CLEAR key is missing, and you can't say DEL or ELSE or PLAY; but you *can* say LPRINT.)

Computers that resemble the IBM PC. If you have a Tandy 1000 or 1000 HD or 1200 HD or 2000 or 2000 HD, read my explanation of the IBM PC. (Before turning on your computer, insert the MS-DOS system master disk. Say BASIC instead of BASICA.)

Computers that resemble the TRS-80 Model 3. If you have a TRS-80 Model 1 or 2 or 12 or 16 or 16B or 100, or a Tandy 200 or 6000 or 6000 HD, or a TRS-80 Pocket Computer 1 or 2 or 3 or 3A or 4 or 5, your computer resembles the TRS-80 Model 3. Here's how to use the TRS-80 Model 3. . . .

The keyboard. Similar to Color Computer 2. You can*not* create the symbols "[", "]", and "\" by holding down the SHIFT key. To create the symbol "[", tap the up-arrow key. To switch to lower case, tap the zero key while holding down the SHIFT key.

Get started, if you don't have a disk. Hold down the BREAK key; and while you keep holding down the BREAK key, flip on the power switch, which is underneath the computer's right side; the computer will say "Cass"; if the writing looks fuzzy, turn the brightness and contrast dials, which are underneath the computer's left side; press the ENTER key; the computer will say "Memory Size"; press the ENTER key again; the computer will say READY.

Get started, if you have a disk. Follow the procedure for the Model 4, with the following exceptions. . . . The disk you insert contains "Model 3 TRSDOS" instead of "Model 4 TRSDOS". After you enter the date (and press the ENTER key), the computer will say "Enter Time"; type the time, by using a 24-hour clock; for example, if the time is 1:45 PM, type 13:45:00 and then press the ENTER key. Then the computer will say "How Many Files"; just press the ENTER key. Then the computer will say "Memory Size"; just press the ENTER key. Then the computer will say READY.

Make a mistake? Same as Model 4.

How to print on paper. Your computer lacks a PRINT key. Instead of pressing a PRINT key, do the following: while depressing the down-arrow key and left SHIFT key simultaneously, tap the colon key.

Your computer lacks an ECHO key. If you're using a disk, you can turn the echo on by typing this—

```
CMD "Z","ON"
```

and turn the echo off by typing this:

```
CMD "Z","OFF"
```

Long variable names. Same as Color Computer 2.

String input. Similar to Color Computer 2. If you're using a disk, the computer says "Out of string space" instead of "?OS ERROR", and says "Out of memory" instead of "?OM ERROR". If you don't say CLEAR, the computer reserves enough memory space to hold just 50 unknown characters.

Disks. Similar to Model 4.
To make the computer print the names of all programs on disk 0, say:

```
CMD "D:0"
```

To make the computer print the names of all programs on disk 1, say:

```
CMD "D:1"
```

Do *not* say FILES or SYSTEM "DIR".
To turn a blank disk into a Radio Shack disk, do *not* say SYSTEM, etc. Instead, put the blank disk into the top drive, and then say:

```
CMD "S"
BACKUP :0 :1
```

The computer will ask, "Source Disk Master Password?" Type the word PASSWORD. The computer will copy all information from the bottom drive's disk to the blank disk. The blank disk will become a Radio Shack disk that's an exact copy of the other disk. The computer will say "TRSDOS Ready". Type the word BASIC. The computer will say "How Many Files". Press the ENTER key. The computer will say "Memory Size". Press the ENTER key. The computer will say READY.

Tapes. Similar to Color Computer 2. The computer will say READY instead of OK, and will print a pair of asterisks instead of an F. The right asterisk flashes.

Editing. Similar to Model 4. When you press the D key, the character will be surrounded by exclamation points instead of backslashes.

Renumbering. To renumber, you must be using disks, and you must say NAME instead of RENUM.

Pausing. To pause (or stop pausing), tap the @ key while holding down the SHIFT key.

Counting to 100. Same as Model 4.

Zones. Each zone is 16 characters wide. Your screen has just 4 zones, so your screen is 64 characters wide.

TAB. Same as Color Computer 2.

LOCATE. Similar to Color Computer 2. Say 64 instead of 32.

Pixels. The X coordinate goes from 0 to 127. The Y coordinate goes from 0 to 47. Say SET instead of PLOT. Your computer doesn't understand LINE, CIRCLE, PAINT, or colors.

Sounds. Your computer doesn't understand BEEP, SOUND, or PLAY.

Random integers. Same as Model 4.

Random decimals. Same as Model 4.

COMPUTERS THAT USE CP/M

The CP/M operating system is used by many computers. The most popular computer using CP/M is the Kaypro 2. Other computers that use CP/M are the Kaypro 2X, the Kaypro 4, the Kaypro 10, and older computers (manufactured by Osborne, Morrow, Cromemco, Vector, North Star, Xerox, and Zorba).

Choosing a language. On CP/M computers, you can use several versions of BASIC, such as MBASIC, SBASIC, CBASIC, and CB-80.

Begin by learning *MBASIC* (which is also called *Microsoft BASIC* and *BASIC-80*), since that version of BASIC is the easiest to learn. Make sure you get version 5 of MBASIC; do not use version 4, which is outdated.

Here's how to use version 5 of MBASIC. . . .

The keyboard. On most CP/M computers (such as the Kaypro), the zero key is Swedish.

The ENTER key says RETURN on it.

Get started. To get started, you need a disk that contains both MBASIC and CP/M.

(When you buy a Kaypro, you get MBASIC on one disk and CP/M on a separate disk. To combine those two disks into a single disk, you must give a SYSGEN command, which is explained in the manual that came with your Kaypro.)

If your computer has just one disk drive, that drive is called *drive A*. If your computer has two disk drives, the drive that's on the top or left is called *drive A*; the other drive is called *drive B*.

(For example, the Kaypro's top drive is called *drive A*; its bottom drive is called *drive B*.)

To start, put the disk that contains both MBASIC and CP/M into drive A. Then turn on the computer's power. (The Kaypro's power switch is at the Kaypro's rear.)

The computer will say:

```
A>
```

Type the word "mbasic", like this:

```
A>mbasic
```

Then press the RETURN key. The computer will say:

```
Ok
```

That procedure works on the Kaypro, but some other computers are different. Instead of "mbasic", some computers require you to type "basic"; and some computers don't require you to type anything at all!

Make a mistake? If the computer gripes at you, press the RETURN key immediately.

After pressing the RETURN key, correct the error, by retyping the line that the computer griped about.

Clearing the screen. The CLS command doesn't work. The method of clearing the screen depends on which computer you have: read the manual that came with your computer.

How to print on paper. Your keyboard lacks an ECHO key. Instead of pressing an ECHO key, tap the P key *while holding down the CONTROL key*.

GO TO. To abort a program, hold down the CONTROL key; and while you keep holding down the CONTROL key, tap the C key.

Disks. If you buy a blank disk and want your computer to use it, you should turn the blank disk into a CP/M MBASIC disk. Here's how to do it, if you have a Kaypro 2 or 2X or 4. (If you have a Kaypro 10 or a different brand of computer, read the manual that came with your computer.)

For a Kaypro 2 or 2X or 4, put the blank disk into drive B. Then type the following:

```
system
copy
c
```

After that final c, press the RETURN key *twice*. The computer will copy all information from drive A's disk to the blank disk. The blank disk will become a CP/M MBASIC disk that's an exact copy of the other disk. After the copying is complete, return to BASIC by typing:

```
mbasic
```

How to choose a name. In the program's name, you must capitalize every letter, like this:

```
save "JOE"
```

Your program's name can be short (such as JOE) or long: up to 8 characters. The name's first character must be a capital letter; the remaining characters should be capital letters or digits.

How to use drive B. If you want JOE to be in drive B instead of drive A, put a B and a colon before JOE, like this:

```
save "B:JOE"
```

Remember to capitalize the B and the JOE.

To make the computer print the names of all the programs in drive B, type this:

```
files "B:*.*"
```

Differences. Instead of saying RENAME "JOE" TO "FRED", say NAME "JOE" AS "FRED". Instead of saying KILL "JOE", say KILL "JOE.BAS", to emphasize that JOE was written in BASIC.

Tapes. Hardly anybody who buys CP/M uses tapes. Therefore, I won't bother explaining CP/M's tape system, which practically nobody uses.

Editing. To edit line 20, type:

```
edit 20
```

If the first character in line 20 is correct (and doesn't need to be edited), press the SPACE bar. If the second character in line 20 is correct also, press the SPACE bar again. Press the SPACE bar for each correct character. Each time you press the SPACE bar, the character that you said was correct appears on the screen.

When you get to the point where the next character on the screen would be *in*correct, press the D key instead of the space bar. The D key means: delete the next character. When you press the D key, the next character will be surrounded by backslashes that mean the character's being deleted.

To insert extra characters in the line, press the space bar several times, until you get to where you want the characters to be inserted. Then press the I key (which tells the computer you want to insert). Then type the characters you want to insert. After typing them, press the key that's marked ESCAPE (or ESC).

When you've finished all the deletions and insertions for that line, press the RETURN key. That makes the screen show the remainder of the line (the right-hand part of the line), and tells the computer you've finished editing the line.

Pausing. Your keyboard lacks a PAUSE key. Instead of pressing a PAUSE key, tap the S key *while holding down the CONTROL key*.

Zones. On a typical CP/M computer (such as a Kaypro), the monitor prints 80 characters per line, and the screen has five zones: zones 1-4 are each 14 characters wide; zone 5 is 24 characters wide.

LOCATE. Your computer doesn't understand the word LOCATE. To find out what to type instead of LOCATE, read the manual that came with your computer.

Pixels. Most CP/M computers can't handle pixels.

Sounds. Most CP/M computers can't produce sounds.

Random integers. Omit the word RANDOMIZE. For a random integer between 1 and 5, say 1+INT(RND*5) instead of RND(5).

WOW! YOU FINISHED APPENDIX 1: VERSIONS OF BASIC!
TURN THE PAGE FOR APPENDIX 2: LISTS.

THEMES

In this volume, a dozen themes appear again and again. Each time a theme reappears, it's analyzed from a different point of view.

Here are those 12 themes, in alphabetical order. Their page numbers are in parentheses.

APPLE

History of Apple (page 18). Apple's disks (36), magazines (63), and computers (80). BASIC for Apple 2c (290), for Macintosh (300), and for other Apples (320).

COMMODORE

History of Commodore (page 16). Commodore's ROM (32), disks (36), magazines (63), and computers (84). BASIC for Commodore 64 (275) and for other Commodore computers (319).

DISKS

Disk hardware (page 34). Survey of disk operating systems (53). Controlling disks from BASIC (136). PC-DOS (192). Controlling disks from Writing Assistant (217), from Word Perfect (219), and from 1-2-3 (243). Data-management systems (250).

EDITING

BACKSPACE key (page 93). Revising BASIC programs in RAM (104), on disk (138), and by using arrow keys (141). Word processing (208). Updating a record using PFS File (255).

FORM LETTERS

Form letters created by INPUT (page 123) and by DATA...READ (148).

GAMES

Computer-generated stories (page 124). Guessing colors (160). Eye test (161). Hearing test (172). Quiz (180). Guessing numbers (187). Coin flipping (188). What your friends want (190). Daily horoscope (191).

GRAPHICS

Graphics printers (page 50). Graphics-input devices (51). Graphics created by BASIC (170) and by 1-2-3 (244).

IBM

IBM maxicomputers (page 12). History of IBM (21). IBM PC's ROM (32) and disks (36). Which IBM PC software to buy (55). IBM PC magazines (63), discount dealers (66), and hardware (72). Variants of the IBM PC (73). PC-DOS (192). BASIC for the IBM PC (270) and for variants (318).

LOOPS

Infinite loops (page 112). READ loops (144). FOR...NEXT loops (154). Advanced loop techniques (177).

PRINTERS

Printer hardware (page 44). Controlling printers from BASIC (111), from PC-DOS (203), from Writing Assistant (217), and from Word Perfect (219). Kinds of printer drivers (222). Using 1-2-3 to print text (243) and graphics (249). Using PFS File to print single records (256) and multiple records (260).

RADIO SHACK

Overview of Radio Shack (page 20). Radio Shack's disks (36), magazines (63), and discount dealers (69). Tandy's IBM clones (76). Color computers (87). Pocket computers (88). Lap computers (88). Z-80 computers (90). BASIC for Color Computer 2 (308), for Model 4 (314), and for other Radio Shack computers (324).

TABLES

Tables created by FOR...NEXT (page 164), by GO TO (179), by 1-2-3 (228), and by PFS Report (264).

COMMANDS

Here are the commands that computers understand. Each list is alphabetized.

MODERN MICROSOFT BASIC

BEEP 172
CIRCLE (100,100),50 170
CLS 110
DATA MEAT,POTATOES 144
DELETE 30-80 107
EDIT 30 141
END 117
FILES 137
FOR X = 1 TO 100 154
GO TO 10 112
IF A$="FINE" THEN PRINT 128
INPUT "WHAT NAME";N$ 122
KILL "JOE" 138
LINE (0,0)-(100,100) 170
LIST 101
LLIST 111
LPRINT "I LOVE YOU 111
LOAD "JOE" 137
LOCATE 3,7 169
NEW 100
NEXT X 154
PAINT (100,101) 170
PLAY "CDG#B-A" 173
PLOT (100,100) 170
PRINT 5+2 95
RANDOMIZE 185
READ A$ 144
RENAME "JOE" TO "FRED" 138
RENUM 142
RESTORE 146
RUN 100
SAVE "JOE" 136
SOUND 440,18.2 172
STOP 117
X=47 118
? 5+2 96
'THIS PROGRAM IS DUMB 143

IBM PC EXTRAS

color 1 273
name "joe" as "ed" 272
pset (100,100) 273
screen 1 273
system 272
width 40 271

APPLE 2C EXTRAS

cat 295
catalog 295
color=11 298
del 30,80 291
delete joe 295
gr 298
hcolor=1 299
hgr 299
hlin 4,8 at 19 298
home 292
hplot 250,140 299
htab 7 297
pr#0 292
pr#1 292
rem Russ has 40 297
speed=0 293
text 298
vlin 20,30 at 4 298
vtab 3 297

MACINTOSH EXTRAS

name "joe" as "ed" 306
pset (100,100) 307
width 60 307

TRS-80 MOD 4 EXTRAS

name "joe" as "ed" 316
print@(3,7), "dry" 317
system 316
system "dir" 316
system "link etc." 315
system "reset *do" 315

COMMODORE 64 EXTRAS

CLOSE 4 276
CLOSE 15 278
CMD 4 276
CONT 280
OPEN 4,4 276
OPEN 15,8,15,etc. 278
PRINT#4 276
REM RUSS HAS 40 280

PC-DOS

b: 201
basic 197
basica 197
c: 201
cd ann 207
chkdsk 196
cls 205
copy con mary 200
date 205
dir 194
diskcopy a: b: 197
echo off 205
erase lambchop 200
format/s b: 198
md ann 206
rd ann 207
rename mary lamb 200
sort 196
sys b: 206
time 205
type mary 200

LOTUS 1-2-3

Copy 235
Data Sort 242
File Erase 243
File Retrieve 243
File Save 243
Graph A 244
Graph B 245
Graph Options 246
Graph Quit 248
Graph Save 249
Graph Type 247
Graph X 246
Graph View 244
Move 234
Print Printer 243
Quit Yes 243
Range Erase 233
Range Format 242
Worksheet Column 238
Worksheet Delete 234
Worksheet Erase 233
Worksheet Global 238
Worksheet Insert 234
Worksheet Titles 239
Worksheet Window 239

BASIC EXAMPLES

Feature	Example	What the computer will do	Page
Arithmetic	`PRINT 5+2`	The computer will print 7.	95
Question mark	`? 5+2`	The computer will print 7.	96
Order of ops	`PRINT 2+3*4`	The computer will print 14.	97
Parentheses	`PRINT 5-(1+1)`	The computer will print 3.	97
String	`PRINT "I LOVE YOU"`	It will print I LOVE YOU.	98
Program	`1 PRINT "I LOVE YOU"` `2 PRINT "YOU TURNED ME ON"`	It will print I LOVE YOU, then print the rest underneath.	100
Colon	`10 PRINT "THINK": PRINT "FLEA"`	Print FLEA under THINK.	107
Semicolon	`10 PRINT "FAT";` `20 PRINT "HER"`	Print FATHER.	108

Blank line	`10 PRINT "JOY"`
	`20 PRINT`
	`30 PRINT "SORROW"`
		Print JOY, then a blank line, then SORROW.	108

Clear screen	`10 CLS`	Clear the screen.	110
Print on paper	`LPRINT "I LOVE YOU"`	Print I LOVE YOU on paper.	111

GO TO	`10 PRINT "CAT"`
	`20 PRINT "DOG"`
	`30 GO TO 10`
		Print CAT, then DOG, then CAT, then DOG, then CAT, then DOG, and so on, forever.	112

Semicolon loop	`10 PRINT "LOVE";` `20 GO TO 10`	Print LOVELOVELOVELOVE etc.	113
Infinite pause	`10 GO TO 10`	Pause until you abort.	116

Skip ahead	`10 PRINT "LET'S MUNCH"`
	`15 GO TO 30`
	`20 PRINT "SANDWICHES UNDER"`
	`30 PRINT "THE TREES"`
		Print just LET'S MUNCH and THE TREES, without printing SANDWICHES UNDER.	117

END	`10 PRINT "SHE SMELLS"`
	`20 END`
	`30 PRINT "OF PERFUME"`
		Print just SHE SMELLS, without printing OF PERFUME.	117

STOP	`10 PRINT "SHE SMELLS"`
	`20 STOP`
	`30 PRINT "OF PERFUME"`
		Print SHE SMELLS, then a message saying it stopped at line 20.	117

Variable	`10 X=47` `20 PRINT X+2`	Print 49.	118
String variable	`10 G$="DOWN"` `20 PRINT G$`	Print DOWN.	120
Long var name	`10 PROFIT.IN.1986 = 3497.18` `20 PRINT PROFIT.IN.1986`	Print 3497.18.	121
INPUT	`10 INPUT "WHAT'S YOUR NAME";N$` `20 PRINT "I LOVE THE NAME ";N$`	Ask for your name, let you reply, then praise your name.	122
IF...THEN	`10 INPUT "HOW ARE YOU";A$` `20 IF A$="FINE" THEN ? "GOOD"`	Ask HOW ARE YOU? If you answer FINE, it will say GOOD.	128

Apostrophe	`10 'THIS PROGRAM IS DUMB`
	`20 C=40' BECAUSE RUSS HAS 40`
	`30 PRINT C`
		Ignore the comments, and print just 40.	143

DATA...READ	`10 DATA MEAT,POTATOES,LETTUCE`
	`20 READ A$`
	`30 PRINT "I LOVE ";A$`
	`40 GO TO 20`
		Print I LOVE MEAT, then I LOVE POTATOES, then I LOVE LETTUCE. Then it will say OUT OF DATA.	144

DATA END	`10 DATA MEAT,POTATOES,LETTUCE`
	`15 DATA END`
	`20 READ A$: IF A$="END" THEN P`
	`RINT "I LOVE THEM ALL": END`
	`30 PRINT "I LOVE ";A$`
	`40 GO TO 20`
		Print I LOVE MEAT, then I LOVE POTATOES, then I LOVE LETTUCE. Then it will print I LOVE THEM ALL.	145

Feature	Example	What the computer will do	Page
RESTORE	10 DATA MEAT,POTATOES,LETTUCE 15 DATA END 20 READ A$: IF A$="END" THEN P RINT "I LOVE THEM ALL": RESTOR E: GO TO 20 30 PRINT "I LOVE ";A$ 40 GO TO 20	Print I LOVE MEAT and I LOVE POTATOES and I LOVE LETTUCE and I LOVE THEM ALL. Then print all that again and again and again, forever.	146
Paired data	10 DATA SAL,SALAD,SUE,SOUP 20 READ P$,F$ 30 PRINT "DEAR ";P$ 40 PRINT " PLEASE BRING ";F$ 50 GO TO 20	Print a letter to SAL, saying PLEASE BRING SALAD. Then print a letter to SUE, saying PLEASE BRING SOUP. Then say OUT OF DATA.	148
DATA search	10 INPUT "WHICH COLOR";C$ 20 DATA WHITE,BLANC,RED,ROUGE 30 READ E$,F$ 40 IF E$<>C$ THEN GO TO 30 50 PRINT "IN FRENCH, IT'S ";F$	Ask WHICH COLOR and let you type a color. Then it will search through the DATA for that color and print its French equivalent.	152
FOR...NEXT	10 FOR X = 1 TO 100 20 PRINT X 30 NEXT X	Print every number from 1 to 100.	154
Multiple copies	10 FOR I = 1 TO 12 20 PRINT "HUSH!" 30 NEXT	Print HUSH twelve times.	157
Finite pause	10 PRINT "YOU SMELL LIKE WINE" 20 FOR I = 1 TO 7000: NEXT 30 PRINT "YOU MUST BE DRUNK"	Print YOU SMELL LIKE WINE, then pause a few seconds, then print YOU MUST BE DRUNK.	160
STEP	10 FOR I = 10 TO 1 STEP -1 20 PRINT I 30 NEXT	Print 10, then 9, then 8, etc., down to 1.	162
Zones	PRINT "SIN","KING"	Print SIN. In zone 2, KING.	163
Table	10 PRINT "NUMBER","SQUARE" 20 FOR I = 3 TO 10 30 PRINT I,I*I 40 NEXT	Print a table, showing the square of each number from 3 to 10.	164
TAB	PRINT TAB(6)"HOT"	Print HOT, indented to spot 6.	165
LOCATE	LOCATE 3,7: PRINT "DROWN"	Print DROWN at line 3's 7th.	169
PLOT	PLOT (100,100)	Draw a dot at pixel (100,100).	170
LINE	LINE (0,0)-(100,100)	Line from (0,0) to (100,100).	170
CIRCLE	CIRCLE (100,100),50	Circle at (100,100), radius 50.	170
PAINT	PAINT (100,101)	Paint, beginning at (100,101).	170
Color	LINE (0,0)-(100,0),2	Draw a green line.	171
Box	LINE (0,0)-(100,100),2,B	Draw a green box.	171
Box fill	LINE (0,0)-(100,100),2,BF	Draw a green box, filled in.	171
BEEP	BEEP	Beep.	172
SOUND	SOUND 440,18.2	Sound at 440 hertz for 1 sec.	172
PLAY	PLAY "CDG#B-A"	Play C, D, G#, B flat, and A.	173
Count from 0	10 C=0 20 PRINT C; 30 C=C+1 40 GO TO 20	Print 0 1 2 3 4 5 6 7 etc.	179
Sum	10 S=0 20 PRINT "NOW THE SUM IS";S 30 INPUT "HOW MUCH TO ADD";X 40 S=S+X 50 GO TO 20	Print NOW THE SUM IS 0. Ask HOW MUCH TO ADD? Wait for you to type a number. Add that number to the sum. Print the new sum. Repeat the process.	182
Rand'm integer	10 RANDOMIZE 20 PRINT RND(5)	Print 1 or 2 or 3 or 4 or 5, chosen at random.	185
Rand'm decimal	10 RANDOMIZE 20 IF RND<.37 THEN PRINT "LOU"	Maybe print LOU; the probability of printing it is 37%.	191

ERROR MESSAGES

If you type a command that the computer can't obey, the computer will gripe, by printing an *error message*.

You can make 6 kinds of errors: syntax errors, numeric errors, printer errors, logic errors, disk errors, and advanced errors. Here are the most popular error messages printed by the IBM PC, in the order you'll probably encounter them. Other computers—by Apple, Commodore, Radio Shack, and the rest of the gang—print error messages that resemble the IBM PC's, though worded slightly differently.

(In parentheses, I'll show you the two-letter abbreviations used by the TRS-80 Color Computer 2.)

SYNTAX ERRORS

If you say PRIND instead of PRINT, the computer will say SYNTAX ERROR (or SN ERROR). That means the computer hasn't the faintest idea of what you're talking about!

If the computer says you have a SYNTAX ERROR, it's usually because you spelled a word wrong, or forgot a word, or used a word the computer doesn't understand. It can also result from wrong punctuation: check your commas, semicolons, and colons. It can also mean you typed a left parenthesis that doesn't have a corresponding right parenthesis, or vice versa.

It can also mean your DATA statement contains a string but your READ statement says to read a number instead. To fix that problem, change the READ statement, by putting a dollar sign at the end of the variable's name.

NUMERIC ERRORS

In a statement such as PRINT 5+2, the + is called the *operation*. The 5 and the 2 are called the *operands*.

If you try to say PRINT 5+2 but forget to type the 2, the computer will say MISSING OPERAND (or MO ERROR).

The biggest number the computer can handle is about 1E38 (which means 1 followed by 38 zeros). If you try to go much higher, the computer will say OVERFLOW (or OV ERROR). For example, if you say 1E39, or you try to multiply together lots of big numbers, you'll get an OVERFLOW ERROR.

If you try to divide by zero, the computer will say DIVISION BY ZERO (or /0 ERROR).

If you feed the computer a number that's inappropriate, the computer will say ILLEGAL FUNCTION CALL (or FC ERROR). For example, if you say DELETE 40, but your program doesn't contain any line whose number is 40, the computer will say ILLEGAL FUNCTION CALL.

PRINTER ERRORS

If your printer runs out of paper, the computer will say OUT OF PAPER.

When the computer sends a message to a device (such as your printer), the computer waits for the device to respond to the message. If the computer has waited a long time without receiving any response from the device, the computer will give up waiting; it will say DEVICE TIMEOUT.

When you try using your printer, if the computer says DEVICE TIMEOUT it means the computer isn't receiving the proper feedback from your printer. That's because your printer isn't properly attached to the computer, or hasn't been turned on, or isn't in a state of being READY.

LOGIC ERRORS

If you say GO TO 40, but your program doesn't contain any line whose number is 40, the computer will say UNDEFINED LINE NUMBER (or UL ERROR). The computer says UNDEFINED LINE NUMBER whenever you try to GO TO a missing line.

The computer handles two *types* of information: numbers and strings. If you feed the computer the wrong type of information—if you feed it a number when you should have fed it a string, or you feed it a string when you should have fed it a number—the computer will say TYPE MISMATCH (or TM ERROR).

When you feed the computer a string, you must put the string in quotation marks, and put a dollar sign after the string's variable. If you forget to type the string's quotation marks or dollar sign, the computer won't realize it's a string; the computer will think you're trying to type a number instead; and if a number would be inappropriate, the computer will say TYPE MISMATCH. So when the computer says TYPE MISMATCH, it usually means you forgot a quotation mark or a dollar sign.

You can make the computer PRINT in two ways. One way is to type a line that begins with the word PRINT: the computer will obey such a line immediately (directly). The other way is to put a number before the word PRINT: the number makes the PRINT statement part of your program, so that the computer won't obey the statement until you type the word RUN. Unlike the word PRINT, the word INPUT *must* have a number in front of it; it *must* be part of your program. If you forget to put a number in front of the word INPUT, the computer will say ILLEGAL DIRECT (or ID ERROR).

DISK ERRORS

To use the disks, you must invent a name for your program. If the name you invent is inappropriate, the computer will say BAD FILE NAME (or FN ERROR).

If you forget to put a disk into the disk drive, or you accidentally leave the disk drive's door open, the computer will say DISK NOT READY. If the disk's surface is scratched or otherwise damaged, the computer will say DISK MEDIA ERROR.

If the disk is write-protected by a write-protect tab (which stops you from writing onto the disk) and you nevertheless try to write information onto the disk, the computer will say DISK WRITE PROTECT (or WP ERROR). If you try to write information onto the disk but the disk doesn't have enough free space to hold the information, the computer will say DISK FULL (or DF ERROR).

The disk's *directory* is a place on the disk that remembers the names of all the disk's programs and files. If the directory is full, so that there's no more room to store names of programs, and you nevertheless try to put another program onto the disk, the computer will say TOO MANY FILES.

If you tell the computer to copy JOE from the disk to the RAM (by saying LOAD "JOE" or RUN "JOE"), but the computer can't find JOE on the disk, the computer will say FILE NOT FOUND (or NON-EXISTENT or NE ERROR).

ADVANCED ERRORS

If you say READ, but the computer can't find any more DATA to read (because the computer has read all the DATA already), the computer will say OUT OF DATA (or OD ERROR).

Whenever you say FOR, you're supposed to say NEXT, and vice versa. If you say FOR without saying NEXT, the computer will say FOR WITHOUT NEXT. If you say NEXT without saying FOR, the computer will say NEXT WITHOUT FOR (or NF ERROR). If your program contains the words FOR and NEXT several times, but you accidentally have more FOR's than NEXT's or more NEXT's than FOR's, the computer will print one of those error messages, because the number of FOR's ought to equal the number of NEXT's.

If your computer doesn't have enough RAM to hold your information, it will say OUT OF MEMORY (or OM ERROR). Part of the RAM is used for handling strings; if that part isn't big enough to hold all your strings, the computer will say OUT OF STRING SPACE (or OS ERROR). If you get one of those error messages, shorten your program (by splitting it into several smaller programs); or (if you're using a TRS-80 Color Computer 2) give a CLEAR command.

SHARE

Share your secret: photocopy the next page, and give it to a friend. If you know a young (or young-at-heart) computerist who's having a birthday, Christmas party, or graduation, turn these volumes into a present: send me the computerist's name, address, and either a greeting card or what you want me to say.

PRICES & POLICIES

The Secret Guide to Computers consists of three volumes. Volume 1 (Secret Skills) covers the fundamentals; volume 2 (Secret Thrills) covers the excitement; volume 3 (Secret Chills) covers the hassles. Each volume costs just $8. Do *not* send money for regular (book-rate) postage and handling: I'll pay for them myself, even if you live in a foreign country. If you're not sure whether to order a book, go ahead: you can return unused books at any time, for a 100% refund.

DISCOUNTS

The list price is $8 per book; so the three-volume set costs $24. To pay less, get together with a friend:

20% discount if you order 2 sets (or any 6 books); you pay just $6.40 per book
30% discount if you order 3 sets (or any 9 books); you pay just $5.60 per book
40% discount if you order 4 sets (or any 12 books); you pay just $4.80 per book
50% discount if you order 50 sets (or any 150 books); you pay just $4 per book
60% discount if you order 600 sets (or 1800 books); you pay just $3.20 per book

START YOUR OWN BOOK BUSINESS

To start your own book business, buy books in quantity from us at discount, and sell them to your friends for slightly under list price. You'll make your friends happy (since they'll pay less than list price), and you'll be happy also (because you'll be helping your friends and also be turning a profit).

IF YOU'RE IN MASSACHUSETTS . . .

If the person who orders the book, or the person we're shipping the book to, is in Massachusetts, we must charge Massachusetts sales tax—unless the book is for a government agency or tax-exempt agency (give us you tax-exempt number) or for resale (give us your Massachusetts resale number) or is a *required text* for a course (tell us which college and course).

PICK-UP

Normally, you get the books by mail. If you live near Boston and don't want to wait for your mailman, come to us and pick up the books yourself. Before you come, phone us at 617-666-2666, so we can give you directions and a guaranteed pick-up time.

PURCHASE ORDERS

If you work for a government agency (such as a public library or a public school), or an accredited private school, or an established bookstore or computer store, or a major computer manufacturer, we'll accept your purchase order, which you must pay 30 days after receiving the books.

REVIEW COPIES

If you're a reviewer for a large magazine, or the head of a large computer department, you might be able to get a complimentary copy of The Secret Guide to Computers for review. To find out, phone us.

CUT OUT THIS COUPON!

How many copies do you want of volume 1 (Secret Skills)? _____

How many copies do you want of volume 2 (Secret Thrills)? _____

How many copies do you want of volume 3 (Secret Chills)? _____

TOTAL number of books: _____

TOTAL number of books	Price per book, usually	Price per book, if you're a Massachusetts resident (including 5% sales tax)
1 - 5	$8.00	$8.40
6 - 8	$6.40	$6.72
9 - 11	$5.60	$5.88
12 - 149	$4.80	$5.04
150 - 1799	$4.00	$4.20
1800 & up	$3.20	$3.36

Multiply your TOTAL number of books by the price per book, and write the TOTAL book price here:

Put a check mark in front of the shipping method you choose. . . .

□ Book rate (or library rate or parcel post or surface), to anywhere in the world: we charge nothing; free!

□ Special handling (which is faster than book rate, and almost as fast as priority mail), to U.S. only: we charge just $1 total, even if your order is large

□ Priority mail (first-class air-mail): we charge $2 *per book* to U.S., $6 *per book* to other countries

□ Express mail (usually overnight), to U.S. only: we charge $10 for the first book, plus $3 for each additional book

FREE brochures on the following topics will become available soon. You can get on our mailing list now, by checking the brochures you want. . . .

□ how you can get later editions of The Secret Guide to Computers

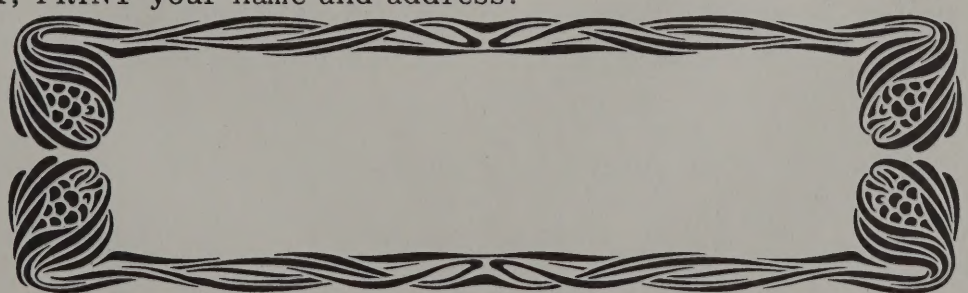
□ how you can get The Secret Guide to Tricky Living

□ how you can attend one-day & two-day courses taught by Russ in various cities

□ how you can attend one-week & multi-week summer courses taught by Russ

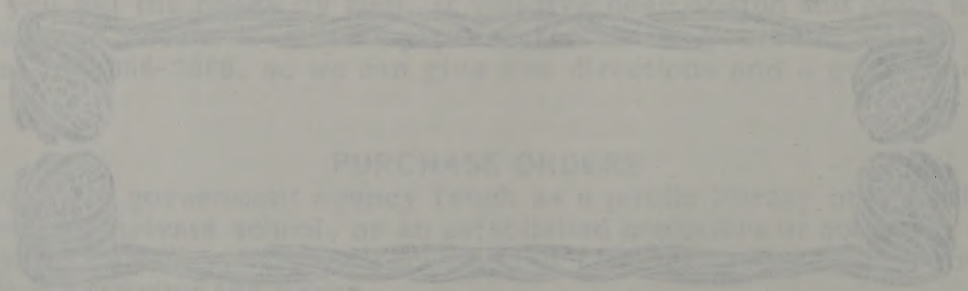
□ how you can get Secret Videotapes

In the box, PRINT your name and address:



On the back, write any comments you have (pro or con) about the Secret Guide.

Add the total book price to the shipping charges (if any); make the check or money order payable to The Secret Guide to Computers. Send to Russ Walter, 22 Ashland St., Somerville, MA 02144.



WHY?

Ten reasons why *The Secret Guide to Computers* has become the most highly praised computer book ever written. . . .

Simple to read: loved by kids & adults who want straightforward talk
Fast-paced: read 10 pages—you'll learn more than from 50 pages of most other texts
Foolproof: carefully tested on beginners & revised to make sure you can't go wrong
Complete: three-volume set covers every aspect of computers & the industry
Inside info: tricks the experts use & anecdotes from programmers' locker rooms
Examples galore: they're easy to type & ready for you to run
Back-up help: call Russ at 617-666-2666, whenever you have computer questions
Ridiculously low price: costs less than any other comprehensive computer text
No math background required, beyond fourth grade: explains all further math
It's fun!

PRAISED BY MAGAZINES

The Secret Guide to Computers has been praised by all the famous microcomputer magazines.

Personal Computing Magazine: "Russ Walter's approach to text-writing sets a new style that other authors might do well to follow. If more college texts were written in the Russ Walter style, more college students would reach their commencement day."

Popular Computing Magazine: "Russ Walter is king of the East Coast computer cognoscenti. The Secret Guide is the first collection of computer writings that one might dare call literature."

Infoworld: "Russ Walter is recognized and respected in many parts of the country as a knowledgeable and effective instructor. People call him at all hours of the night from as far away as Australia for advice."

Computer Update: "Russ Walter is Boston's computer guru. He's years ahead of the pack that claims to have ways of instructing computer novices."

Eighty Micro: "Theatrical, madcap Russ Walter is a cult hero. Lots of books teach novices the difference between RAM and ROM, but Walter is more ambitious and more irreverent."

Interface Age: "How did Russ Walter do it? It's a best buy."

Softalk: "The material is culled from special seminars given by Russ Walter, New England's computer guru. He pulls no punches, and fires well-deserved salvos at many sacred cows."

Abacus: "Alternative-culture Walter provides the best current treatment of programming languages. Walter takes pride in challenging and shocking the staid and comfortable."

Connecticut Computer Society News: "Russ Walter's books have been used by insiders for years. He's widely known as Boston's computer guru."

Scientific American: "It's irresistable. No room that holds a small computer and an adult learning to use it is well equipped without the Guide."

The Whole Earth Catalog, in its Coevolution Quarterly update: "The Secret Guide is the only introductory survey of equipment that's kept up to date. The book tells the bald truth in comprehensible language."

The Detroit News: "Russ Walter is a legendary teacher of personal computers. Although lots of books claim to be comprehensive, his Secret Guide is the only one that really includes everything."

The Australian, one of Australia's major periodicals: "Read the Secret Guide and you'll have a very good idea of how computers and programs work—and a good laugh as well. The book's coverage of programming is intelligent, urbane, and extremely funny."

Cider Press: "The Guide is as difficult to lay aside as a first-rate mystery. The Guide should be given to all beginners with the purchase of their computers."

Computer Bargain Info: "Russ Walter is Boston's humble computer guru. His Secret Guide is widely acclaimed by experts as brilliant."

Many other periodicals have offered similar praises. Copies of the reviews are on pages 2-4.